

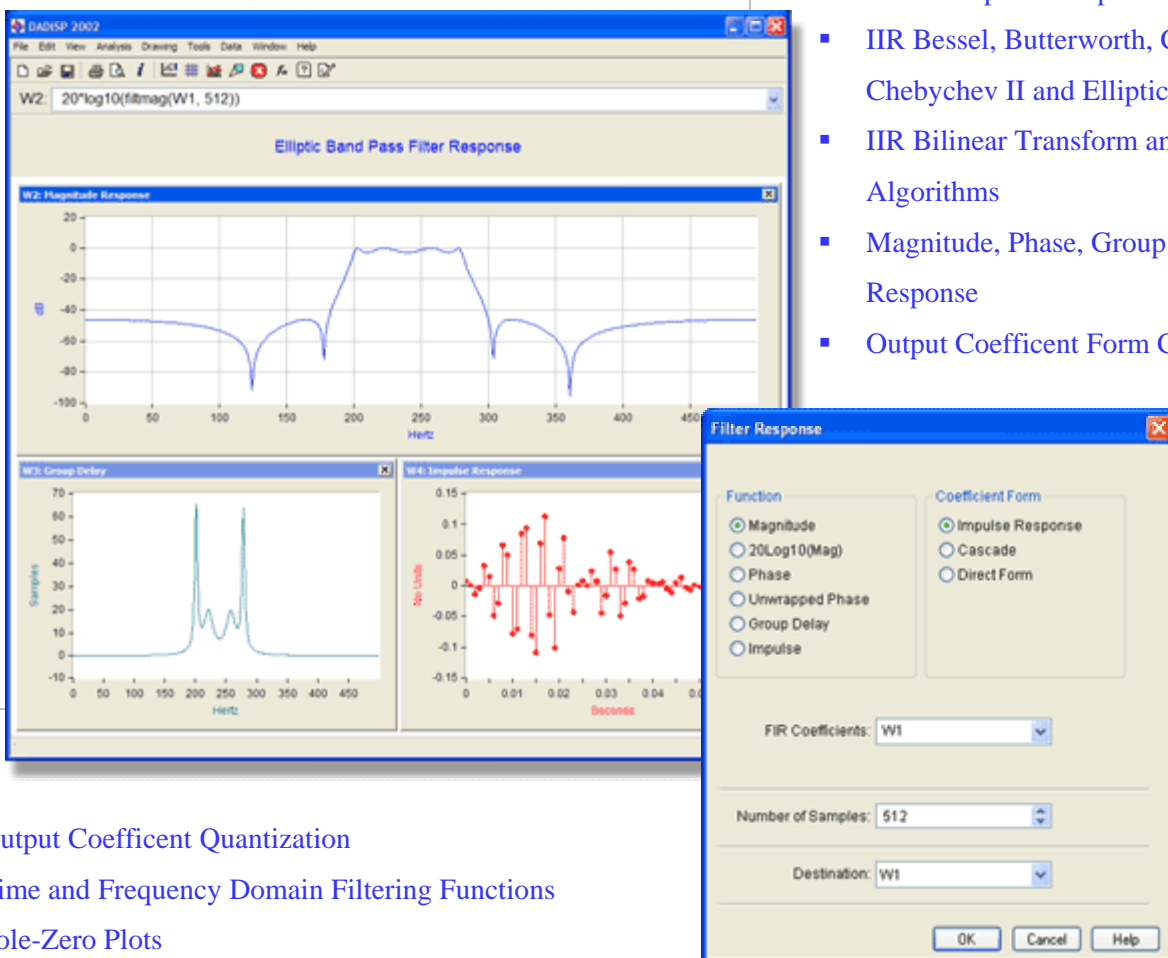
# DADiSP / Filters

## Digital Filter Design Module

DADiSP/Filters is a menu-driven module for FIR and IIR digital filter design and analysis. From any [DADiSP Worksheet](#), you can quickly design, view and analyze both FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filters. Once you have designed a filter, you may filter the raw data, then view and analyze the filtered signal. Through the easy-to-use dialog boxes or simple one line functions, you can tune the filter iteratively, re-filtering the data until you have separated the signal from the noise cleanly.

### KEY FEATURES

- Simple User Interface
- Lowpass, Highpass, Bandpass, Bandstop and Multiband Filters
- Finite Impulse Response (FIR) Filter Design
- FIR Hilbert Transforms and Differentiators
- FIR Remez Exchange and Kaiser Window Design Algorithms
- Infinite Impulse Response (IIR) Filter Design
- IIR Bessel, Butterworth, Chebychev I, Chebychev II and Elliptic Filters
- IIR Bilinear Transform and Matched Z Design Algorithms
- Magnitude, Phase, Group Delay and Impulse Response
- Output Coefficient Form Conversion



- Output Coefficient Quantization
- Time and Frequency Domain Filtering Functions
- Pole-Zero Plots

## New Features

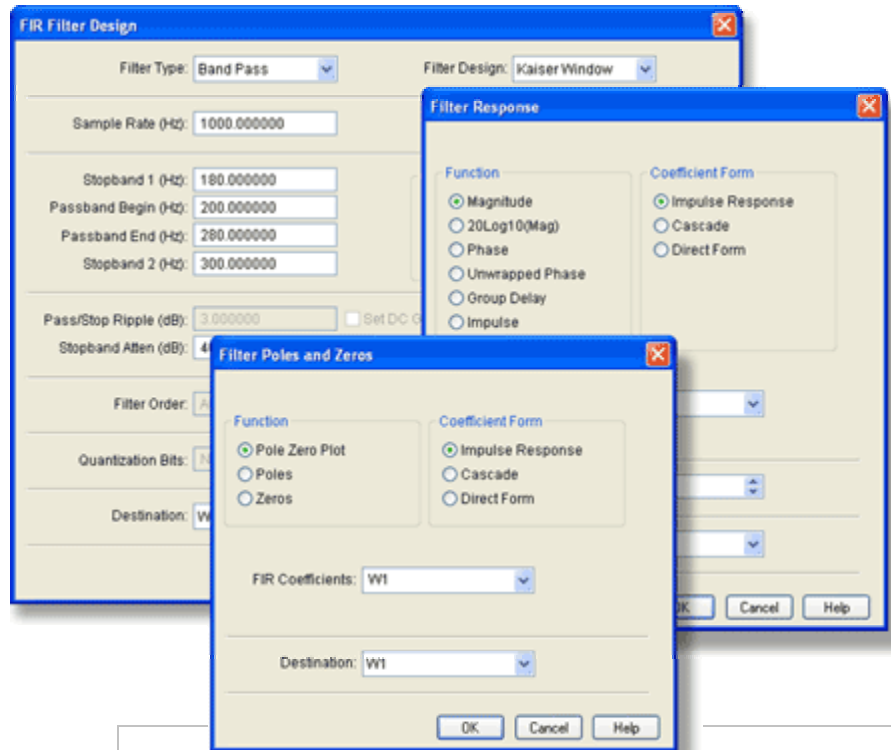
DADiSP/Filters Version 5.0 includes a completely redesigned user interface to streamline the process of designing and applying digital filters. Straightforward dialog boxes with automatic option validation simplifies both the design and analysis of filters.

Filter coefficients can be easily converted to various filter structures and quantization routines are included to help simulate DSP chipsets.

IIR Bessel filters and the Matched Z Transform design method have been added. Linear phase FIR Kaiser filters have been expanded and enhanced. Both time and frequency domain filtering routines have been optimized to provide more efficient filter processing.

## Digital Filter Design Module

DADiSP/Filters is a menu-driven, digital filtering module that adds complete FIR and IIR filtering capabilities to DADiSP. DADiSP/Filters allows you to quickly design, analyze and process both FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) digital filters from easy-to-use dialog boxes or simple one line functions. On-line help and examples are also provided.



## FILTERS 5.0 NEW FEATURES SUMMARY

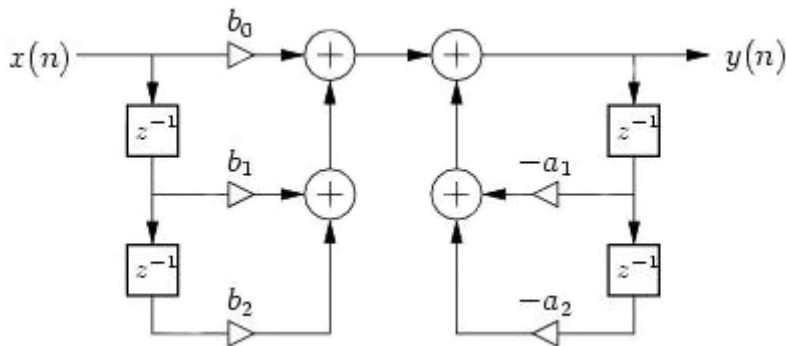
- Streamlined Interface
- IIR Bessel Filters
- IIR Matched Z Design Algorithm
- Improved FIR Kaiser Window Filters
- Coefficient Conversion and Quantization
- Optimized Filter Processing Functions

## Design, Analyze and Apply

DADiSP/Filters gives you the power to easily build digital filters that emulate hardware based designs for testing and verification or perform filtering operations not possible with traditional analog methods. DADiSP/Filters allows you to remove noise generated during the data collection process. DADiSP/Filters is the perfect complement to [GPIBLab](#), DADiSP's acquisition module for collecting data from IEEE-488 based instruments and [DADiSP/AdvDSP](#), a sophisticated module for advanced signal processing techniques.

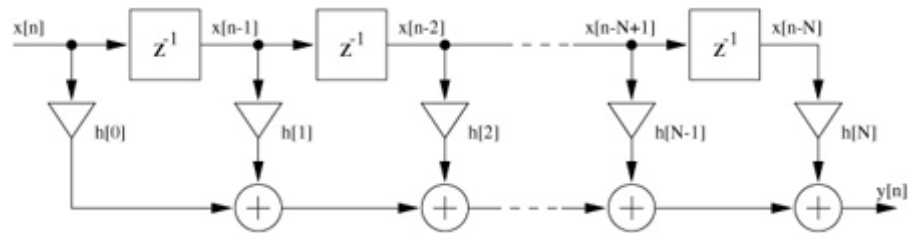
## Coefficient Conversion

Filter coefficients can be converted to and from Cascade, Direct and FIR form and the coefficients can be quantized to emulate DSP chipsets.



## FIR Filters

The FIR module creates linear phase lowpass, highpass, bandpass, bandstop, multiband, Hilbert transformers and differentiators using the Parks-McClellan/Remez Exchange optimal design algorithm. The filter order can be specified or automatically estimated from the particular filter specifications. The Kaiser Window method is also provided, capable of creating both very high order FIR filters and extremely tight, narrowband filters.



## IIR Filters

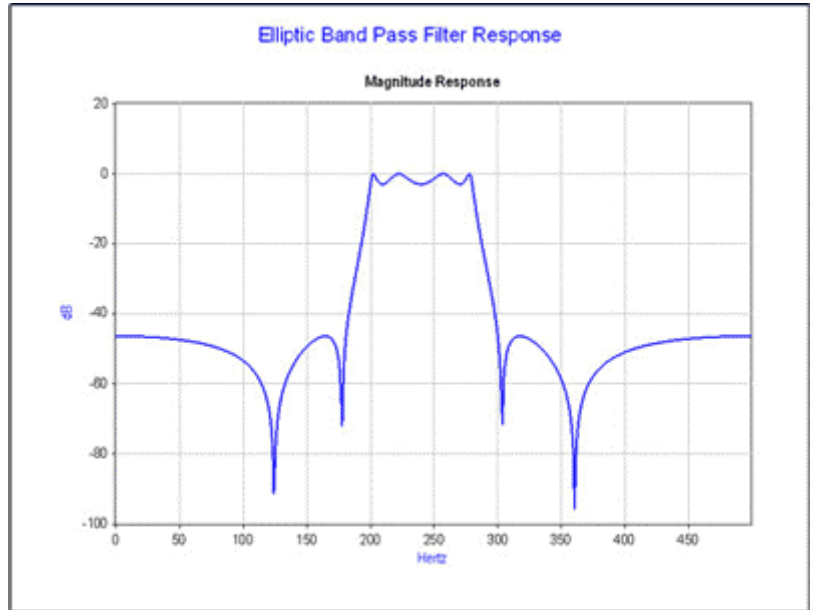
The IIR module supports Bessel, Butterworth, Chebychev I, Chebychev II and Elliptic designs for lowpass, highpass, bandpass and bandstop recursive filters. The Bilinear transform method is employed and the resulting IIR coefficients can be determined in optimal multistage cascade format or traditional Direct form. The module also supports the Matched-Z transform method to approximate linear phase characteristics for IIR Bessel filters.

## Filters Response

The impulse, magnitude, phase and group delay characteristics can be calculated for any filter. Pole-Zero plots are also supported. Efficient time and frequency domain filtering algorithms are provided to apply the filter to any series.

## Fully Integrated

The filters module is fully integrated with DADiSP to provide a complete digital filter design, analysis, display and processing environment. The DADiSP/Filters user interface is dialog based, eliminating the need to memorize argument lists or formulas and allows quick recall of previous filter designs. The filter coefficients are automatically displayed in a DADiSP window and can be saved for further use by DADiSP or other programs.



## Filter Functions

Although most users access DADiSP/Filters through the dialog based interface, DADiSP/Filters includes over standalone 50 functions. The following table is a summary of each function.

\* Indicates new or improved Version 5.0 functions.

### FIR Filters Functions

bandpass	Designs a FIR linear phase bandpass filter
bandstop	Designs a FIR linear phase bandstop filter
diff	Designs a FIR differentiator
fastfilter	FFT based FIR filtering
highpass	Designs a FIR linear phase highpass filter
hilbert	Designs a FIR Hilbert transformer
kwbpass*	Designs a Kaiser window FIR bandpass filter
kwbstop*	Designs a Kaiser window FIR bandstop filter
kwhpass*	Designs a Kaiser window FIR highpass filter
kwlpass*	Designs a Kaiser window FIR lowpass filter
lowpass	Designs a FIR linear phase lowpass filter
remez	Creates multiband FIR linear phase filters

### Filter Coefficient Conversion Functions

cas2dir*	Converts Cascade form to Direct form
dir2cas*	Converts Direct form to Cascade form
fir2dir*	Converts FIR impulse form to Direct form
fir2cas*	Converts FIR impulse form to Cascade form

### IIR Filters Functions

bessel*	Designs an IIR Bessel filter
butterworth	Designs an IIR Butterworth filter
cascade	Filters a time domain input with an IIR filter
cheby1	Designs an IIR Chebychev I filter
cheby2	Designs an IIR Chebychev II filter
elliptic	Designs an IIR Elliptical filter

### Filter Response Functions

filtgrpdelay*	Calculates group delay of any filter
filtmag*	Calculates any filter magnitude response
filtimp*	Calculates any filter impulse response
filtphase*	Calculates any filter phase response
firmag	Calculates FIR filter magnitude response
firphase	Calculates FIR filter phase response
iirimp*	Calculates IIR filter impulse response
iirmag*	Calculates IIR filter magnitude response
iirphase*	Calculates IIR filter phase response
filtgrpdelay*	Calculates group delay of any filter
filtmag*	Calculates any filter magnitude response
filtimp*	Calculates any filter impulse response

## Filtering Functions

<code>dirfilter*</code>	Apply Direct form filter in the time domain
<code>dirfilterF*</code>	Apply Direct form filter in the frequency domain
<code>filtdataF*</code>	Apply any filter in the frequency domain
<code>filtdata*</code>	Apply any filter in the time domain
<code>firfilterF*</code>	Apply FIR filter in the frequency domain
<code>firfilter</code>	Apply FIR filter in the time domain
<code>iirfilterF*</code>	Apply IIR filter in the frequency domain
<code>iirfilter</code>	Apply IIR filter in the time domain
<code>dirfilter*</code>	Apply Direct form filter in the time domain
<code>dirfilterF*</code>	Apply Direct form filter in the frequency domain
<code>filtdataF*</code>	Apply any filter in the frequency domain
<code>filtdata*</code>	Apply any filter in the time domain

## Misc Filtering Functions

<code>filtzeros*</code>	Calculates zeros of any filter
<code>filtpoles*</code>	Calculates poles of any filter
<code>fir</code>	Evaluates a FIR difference equation
<code>firpz</code>	Creates an FIR filter zero plot
<code>firzeros</code>	Calculates zeros of an FIR filter
<code>fullfir</code>	Converts FIR filter to full band linear phase
<code>iir</code>	Evaluates an IIR difference equation
<code>iirgrpdelay*</code>	Calculates group delay of an IIR filter
<code>iirpoles</code>	Calculates poles of an IIR filter
<code>iirpz</code>	Creates an IIR filter pole-zero plot
<code>iirzeros</code>	Calculates zeros of an IIR filter
<code>polecoef</code>	Converts IIR biquad to direct pole coeff form
<code>quantize*</code>	Quantize filter coefficients to N bits
<code>zerocoeff</code>	Converts IIR biquad to direct zero coeff form
<code>unwrap</code>	Phase unwrapping using Schafer's algorithm