ASN Filter Designer

Filter designer GUI user's guide

March 2017
ASN15-DOC001, Rev. 10

For public release

Legal notices

Advanced Solutions Nederland B.V.

www.advsolned.com

support@advsolned.com

Technical documentation feedback

If you would like to make a suggestion or report an error in our documentation, please feel free to contact us. You are kindly requested to provide as much information as possible, including a full description of the error or suggestion, the page number and the document number/description. All suggestions or errors may be sent to: documentation@advsolned.com

1. Introduction

Thank you for your interest in the ASN Filter Designer. This product is available in three flavours, as summarised below:

Licence type	Demo	Educational	Professional
Max IIR filter order (*design method*)	6	20	100
Max FIR filter order (*design method*)	64	200	499
Max num poles/zeros (*design method + script*)	100	200	500
FIR Multiband	4 bands	5 bands	8 bands
Save project	✗	✓	✓
Save analyser data	✗	✓	✓
Export to Excel	✗	✓	✓
Export charts	✗	✓	✓
Max interface variables (ASN FilterScript)	3	6	20
Licence	non-commercial use	non-commercial use	commercial use

- All licences are perpetual licences.

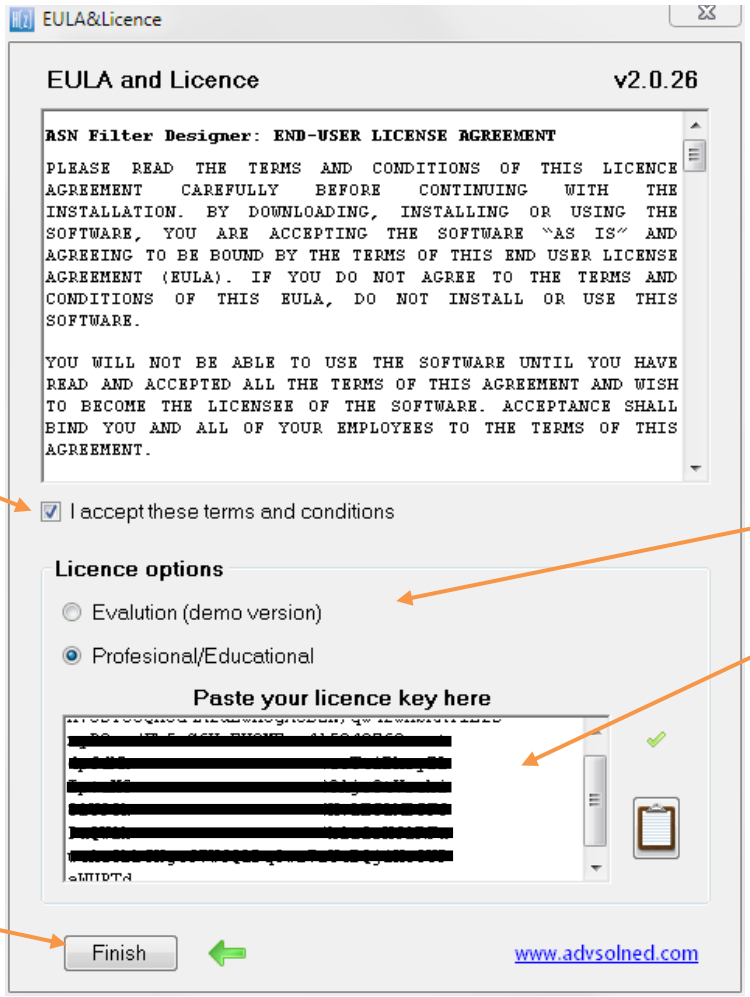
Please contact us at sales@advsolned.com if you would like to buy a licence for the educational or professional edition. We also offer academic and group licensing discounts, so please ask!

1.1. Getting started

The ASN filter designer has been designed around Microsoft's .NET technology, and as such requires .NET framework 4 to be installed before installation of the ASN Filter designer can continue. Where, the currently supported operating systems are: Windows 10, Windows 8, Windows 7, Windows Vista and Windows XP Service Pack 3.

After downloading the '.zip' archive from ASN's website, please double click on `setup.exe` in order begin the installation procedure. `setup.exe` will automatically check to see if the .NET framework needs to be downloaded (for most versions of Windows this will not be necessary), and will then continue with the installation.

1.1.1. EULA



STEP1: Check the T&Cs

STEP2: Choose the licence model that you would like to use.

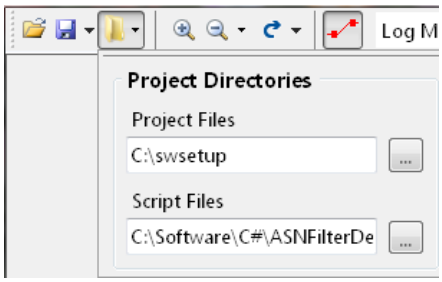
Enter your licence key

STEP3: Click finish and begin using the tool.

Your licence details may be viewed inside the GUI via the **Help > About** menu.

1.1.2. User directories

After finishing the installation, it is advisable to set up your project directories:



Project Files: This is the default location of where all design project files are stored. The tool is shipped with several example project files, but you may modify this location to suit your needs.

Script Files: This is the default location of all Filter Script project files (see section 8.2.3 for more information).

These settings will be automatically saved when the GUI closes.

If you are evaluating the software, this step may be skipped.

1.1.2.1. Other important directories

Directory name	Description
\Datafiles	Default location for external data files to be loaded into the signal analyser.
\Matlab	Matlab software development framework.
\Scilab	Scilab software development framework.
\ANSIC	ANSI C software development framework.

The software development frameworks allow users to quickly and easily import and integrate filters designed within the ASN filter designer into 3rd party applications, such as an algorithm within Matlab. The software frameworks are discussed in section 5.2.4.

1.1.3. Computer requirements

Processor: The high performance DSP libraries are based around Intel's MKL technology, which requires an Intel processor in order to achieve optimal performance. Although the tool will run on other types of processors, the performance will not be optimal and may in some cases lead to sluggish performance. Therefore, an Intel processor with a system passmark benchmark of at least 1500 is recommended.

Please see http://www.cpubenchmark.net for more information.

Screen: A screen size of at least 14 inches is recommended, but the UI will be automatically scaled for smaller screen sizes.

Mouse: any windows compatible mouse with a mouse wheel (required for zooming).

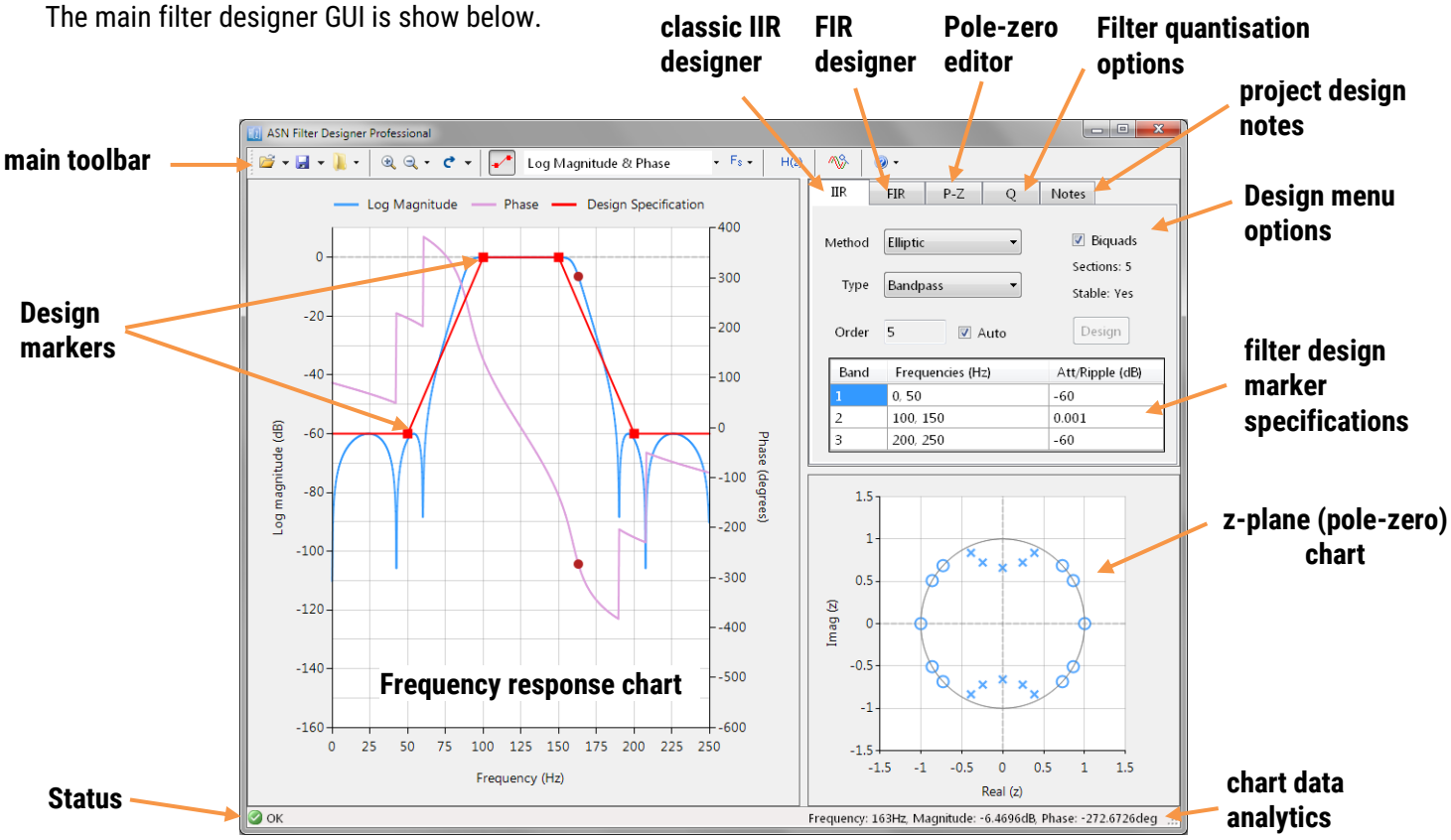
1.1.4. Technical references

This user's guide is intended as a concise reference guide, and assumes that the reader has a firm grasp of signal processing techniques. For any readers looking for background material, please consult the following references:

- Digital signal processing: principles, algorithms and applications, J.Proakis and D.Manoloakis
- Digital signal processing: a practical approach, E.Ifeachor and B.Jervis.
- Digital signal processing and signal processing, L.Jackson.
- Understanding digital signal processing, R. Lyons.

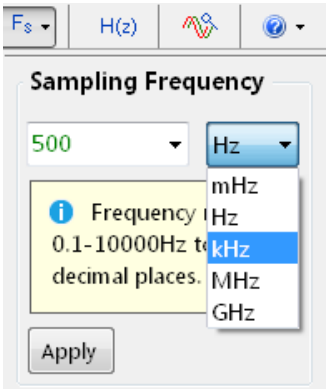
2. The filter designer GUI

The main filter designer GUI is show below.



2.1. Setting the sampling frequency (Fs)

Before embarking upon any design, it is recommended to set the **Sampling Frequency, Fs**. Note that the specified sampling frequency is used for all filters and the signal generator.



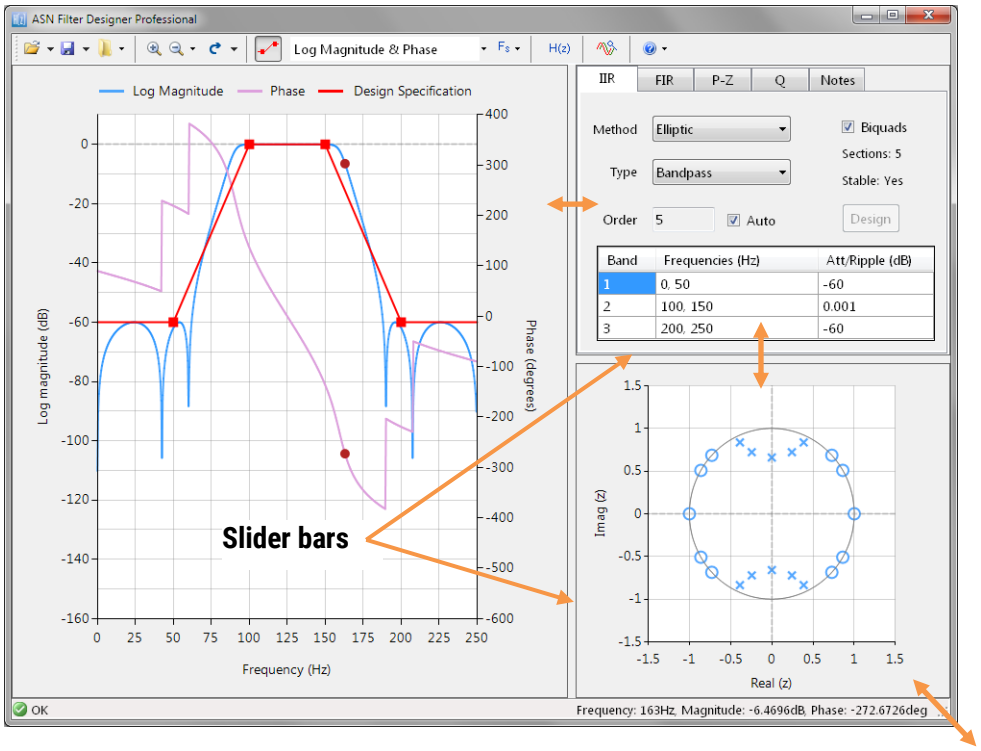
The information textbox will offer advice regarding valid frequency ranges.

The sampling frequency may be specified up to 4 decimal places, which is useful for designing filters based on fractional sampling frequencies, such as multiples of the 44.1kHz audio standard. Common examples include audio interpolation filters: 44.1kHz × 128 = 5.6448MHz and 44.1kHz × 256 = 11.2896MHz.

Changing the sampling frequency will delete all poles and zeros and reset the design to its default settings. Therefore, ensure that *you set the correct sampling frequency* before customising your design!

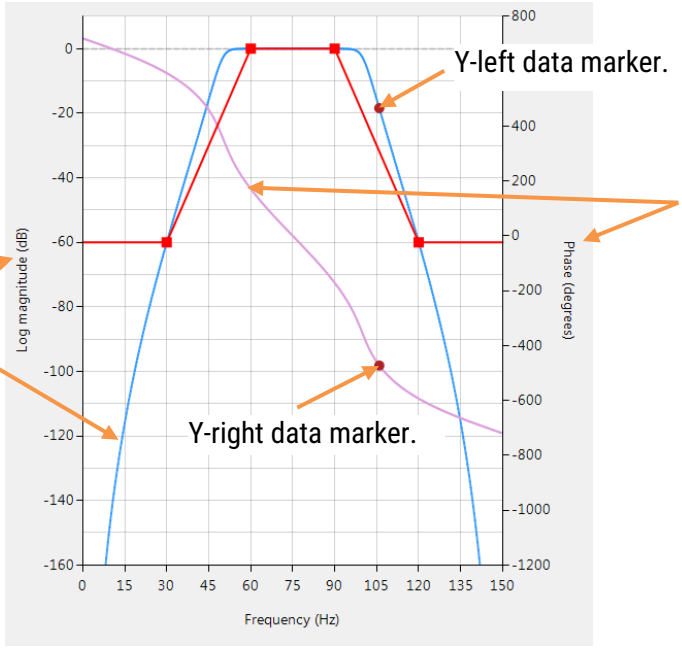
2.2. Resizing the charts

Use the slider bars to resize the design menu area, the P-Z (pole-zero) chart or the frequency response chart.



2.3. Frequency response chart

The frequency response chart shows the designed filter's frequency response, data markers (used for the chart data analytics) as well as the design markers used for the design specification.

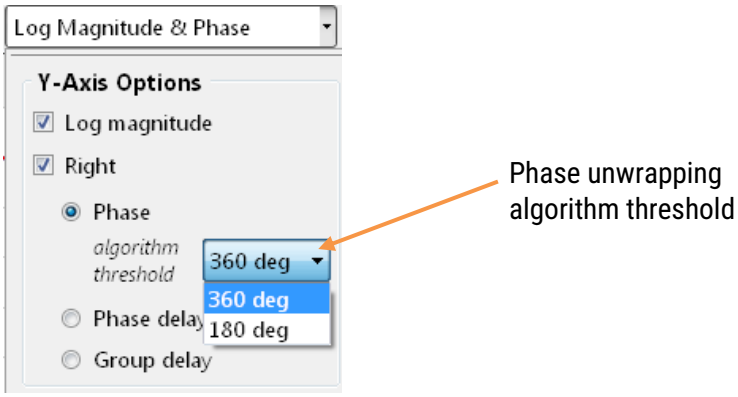


Y-left (blue line):
- log magnitude.

Y-Right (purple line):
- unwrapped phase.
- group delay.
- phase delay.

2.3.1. Changing chart view

Select which data you wish to view via the **main toolbar** > **chart options** menu, as shown below:



As seen, the left Y-axis is always log magnitude, and the right Y-axis may be switched between phase, phase delay and group delay respectively.

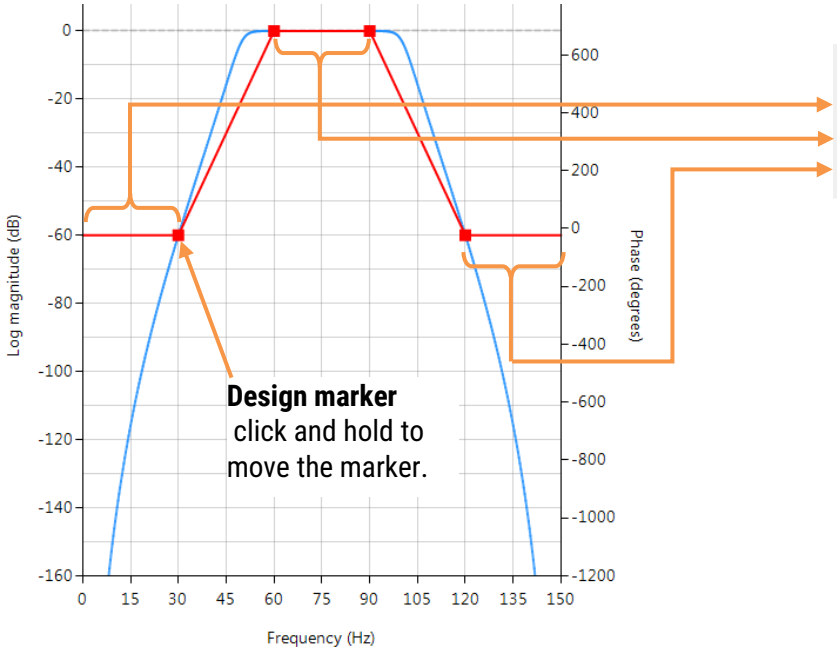
The phase unwrapping algorithm threshold allows you to switch between **360 degrees** (default) and **180 degrees**. Where the latter is particularly useful for viewing the continuous phase spectrum.

2.3.2. Design specification markers

The design specification markers concept forms the essence of the intuitiveness of the tool, allowing designers to graphically specify their design specifications and see the true filter frequency response in real-time:

Click and hold the red square with the left mouse button and then drag in any direction to modify the marker's position. The filter specification table will automatically be updated.

The filter specification is broken up into *bands* and summarised in the filter specification table.



Band	Frequencies (Hz)	Att/Ripple (dB)
1	0, 30	-60
2	60, 90	0.001
3	120, 150	-60

You may fine tune the design specification by manually modifying the entries in the filter specification table.
NB. This will also change the appearance of the design specification line to dashed lines ---- until the **Design** *button is pressed (see overleaf for more details).*

2.3.2.1. Design specification line appearance

The design specification line appearance may take one of three settings depending on the action being performed:

Normal: This is the standard setting used for designing with the IIR and FIR design methods.

Dashed: If a filter specification table entry is being modified by the user or in the quantisation menu.

Illustration: If using the P-Z editor (**User Defined** mode), and an H1 filter designed by the design methods is present.

You may show or hide the design specification line via  the button in the main toolbar.

2.3.3. Exporting charts

Licensed users may export  both the frequency response and z-plane chart to **clipboard** or as a high resolution picture file – where, **bmp, gif, jpeg, emf** and **png** formats are supported. You may also export the chart data to a text file (**Text file**), which allows for further customisation (such as adjusting line thickness and changing the chart axes titles) in third party programs.

2.3.4. Data analysis

Data analysis is performed with the mouse. Where, moving the mouse over the chart will automatically produce data markers and data analytics (shown at the bottom right side of the GUI).

The data analysis algorithm implements a specialised version of the Discrete Fourier transform, which allows designers to perform high resolution frequency analysis of any point of interest on the magnitude, phase, group delay and phase delay charts respectively.

Care should be exercised when analysing the frequency response chart, as the specialised implementation computes the Fourier component at specified frequency points rather than the standard $\frac{F_S}{N}$. The virtue of this implementation results in the evaluation of the exact magnitude and phase values at specific frequency points. Although this is desirable for the magnitude response (allowing you to see the exact magnitude at a given frequency point), the unwrapped phase estimate may vary slightly when panning and zooming over certain ranges - see section 2.3.7 for more information.

2.3.4.1. Zooming in/out

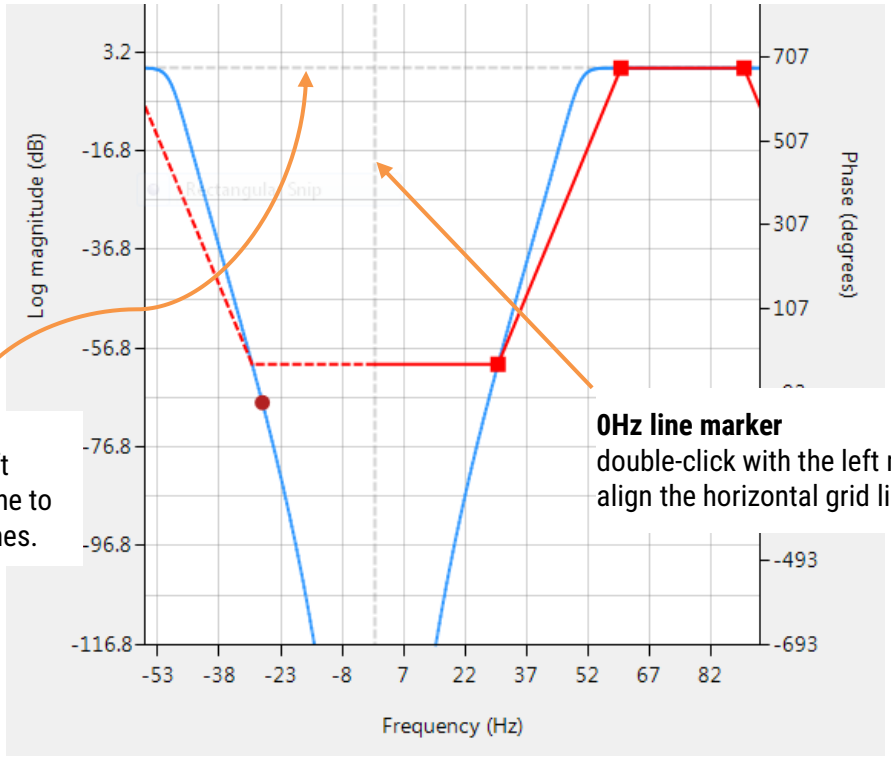
You may zoom in and out into any area by using the mouse wheel:

The zoom is centred on the position of the mouse pointer, in order to accommodate regional zoom functionality.

2.3.4.2. Panning

Panning may be achieved by depressing the left mouse button and dragging the mouse in any direction. Where, the frequency panning range is limited to ± Nyquist.

2.3.5. The 0dB and 0Hz line markers

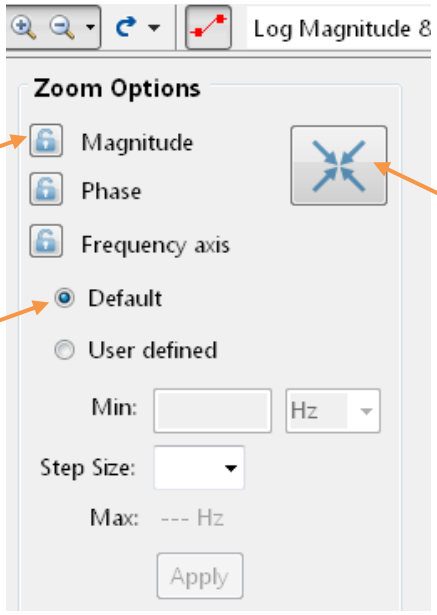


0dB line marker
double-click with the left mouse button to align the to align the vertical grid lines.

0Hz line marker
double-click with the left mouse button to align the horizontal grid lines.

2.3.6. Chart zoom options

The ASN filter designer provides designers with a comprehensive zooming menu for undertaking analysis of demanding filter designs.



Lock/unlock magnitude (y-left) axis.

reset zoom on all axes.

Default zoom allows zoom with the mouse wheel over the frequency range ±Nyquist (±Fs/2).

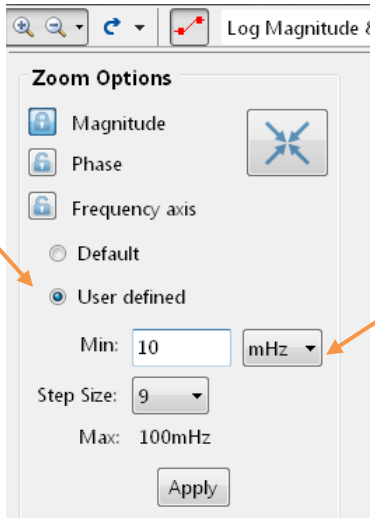
2.3.6.1. Locking axes

In order to simplify data analysis with three axes, you may lock a specified axis for zooming/panning purposes. This has the advantage of allowing you to customise each chart axis to your exact requirements.

2.3.6.2. Zooming to a specific frequency range

user defined zoom
You may zoom to specific frequency range with the **User defined** zoom function.

The universality of this function allows you zoom to mHz resolution even when the sampling rate is in the MHz region!



Choose the frequency scale that you wish to zoom to. Notice here that we are setting the x-axis to the range: `10-110mHz`.

Panning is disabled on the x-axis (frequency) when this function is enabled!

2.3.7. Interpreting the phase spectrum and 0Hz

A specialised implementation of the DFT (discrete Fourier transform) allows designers to analyse any frequency range desired. This feature allows designers to perform detailed phase discontinuities analysis as well as detailed data analysis at other frequency scales (such as mHz, Hz, kHz etc) even when the sampling rate is in MHz range.

Care should be exercised when interpreting the results, as only 500 DFT data points are used over the desired frequency range, instead of the approach adopted by other tools that use thousands of DFT points. As a result, zooming or panning over certain ranges may give slightly different results to the original ±Nyquist (±Fs/2) phase plot. The difference is attributed to the higher resolution (finer step size) between the DFT computation points, which affect the phase unwrapping algorithm which is a relative function (the magnitude spectrum estimates will always remain the same). These zoomed values should be interpreted as the true phase values.

In order to overcome glitches, the DFT is not actually computed at 0Hz, but at 1e-4Hz. Where, this minimum value is automatically adjusted depending on the frequency scale.

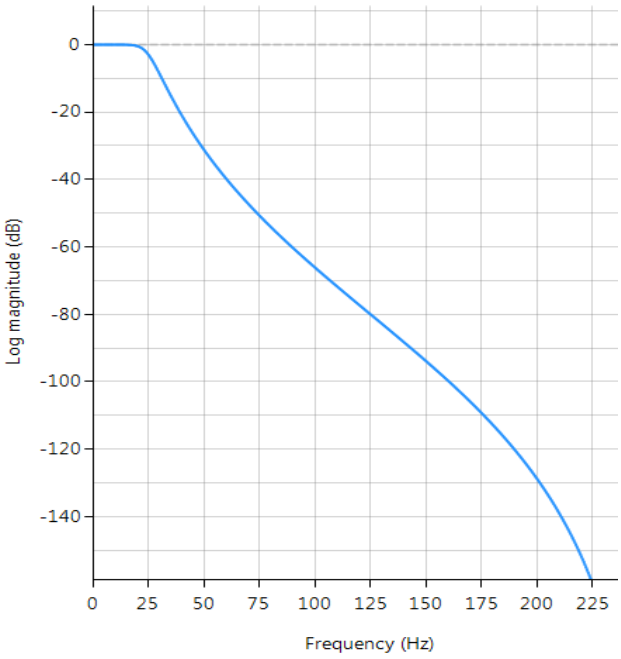
2.3.8. Errors in high order polynomials

The tool will for FIR filters and the filter script use the given **Num** and **Den** polynomials for computation. However, if these positions are modified via the P-Z editor, they will be handled via the roots-to-poly algorithm which will attempt to reconstruct the polynomial from the presented roots using double precision arithmetic. For lower orders this will generally result in an almost identical polynomial, but as a consequence of the errors inherent to the root finding algorithm, higher order polynomials (> 60 or so) may significantly deviate from the ideal result.

2.4. Classical IIR Filter design

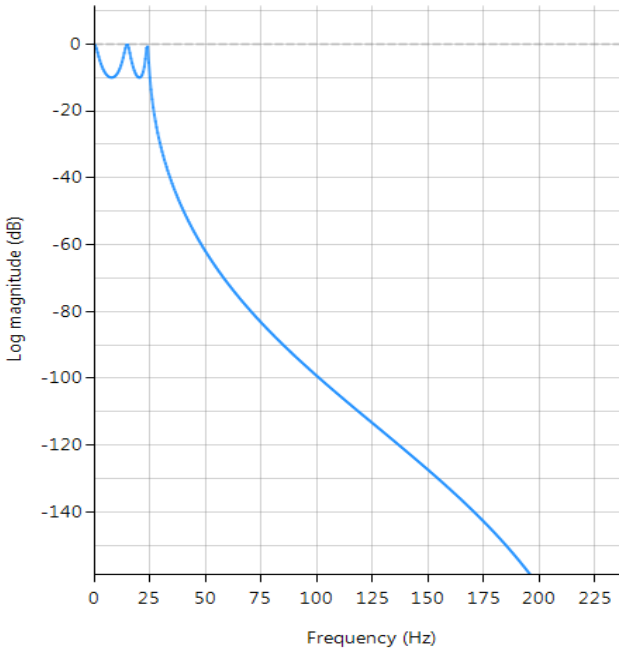
The IIR filter designer allows developers to implement the following classical design prototype methods for lowpass, highpass, bandpass and bandstop filters:

Butterworth (5th order)



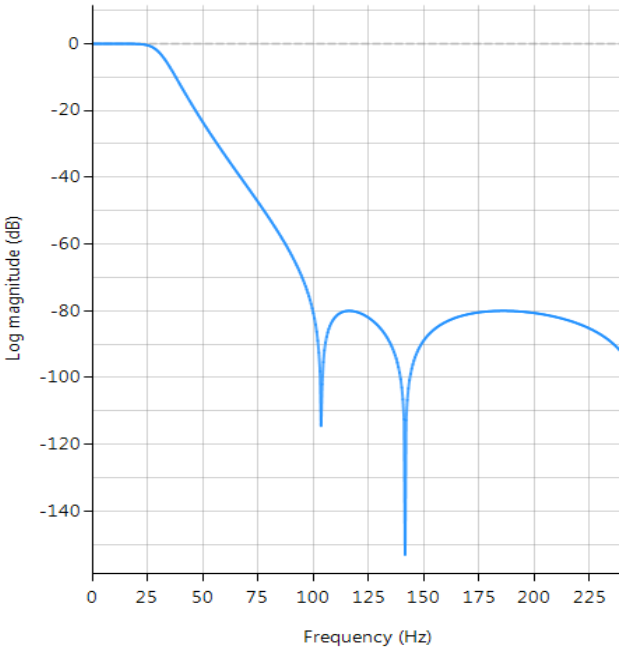
- Smooth monotonic response (no ripple).
- Slowest roll-off for equivalent order.
- Highest order of all supported prototypes.

Chebyshev type I (5th order)



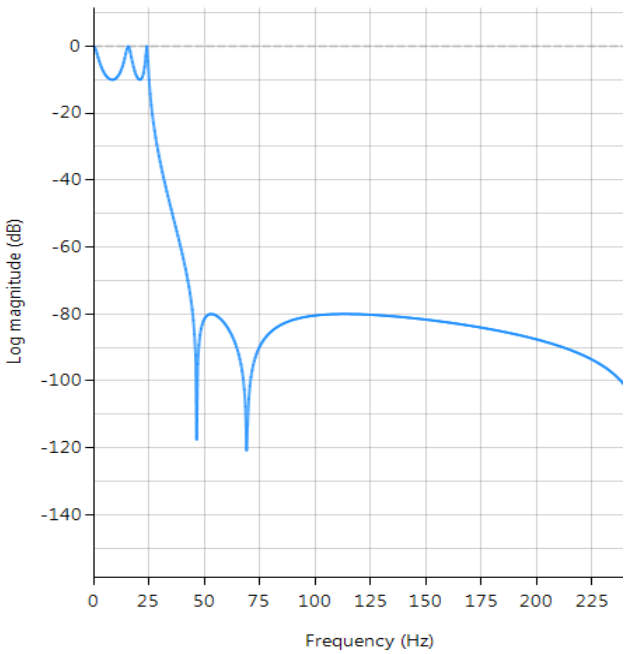
- Maximally flat stopband.
- Faster roll off (passband to stopband transition) than Butterworth.

Chebyshev type II (5th order)



- Maximally flat passband.
- Slower roll off (passband to stopband transition) than Chebyshev Type I.

Elliptic (5th order)

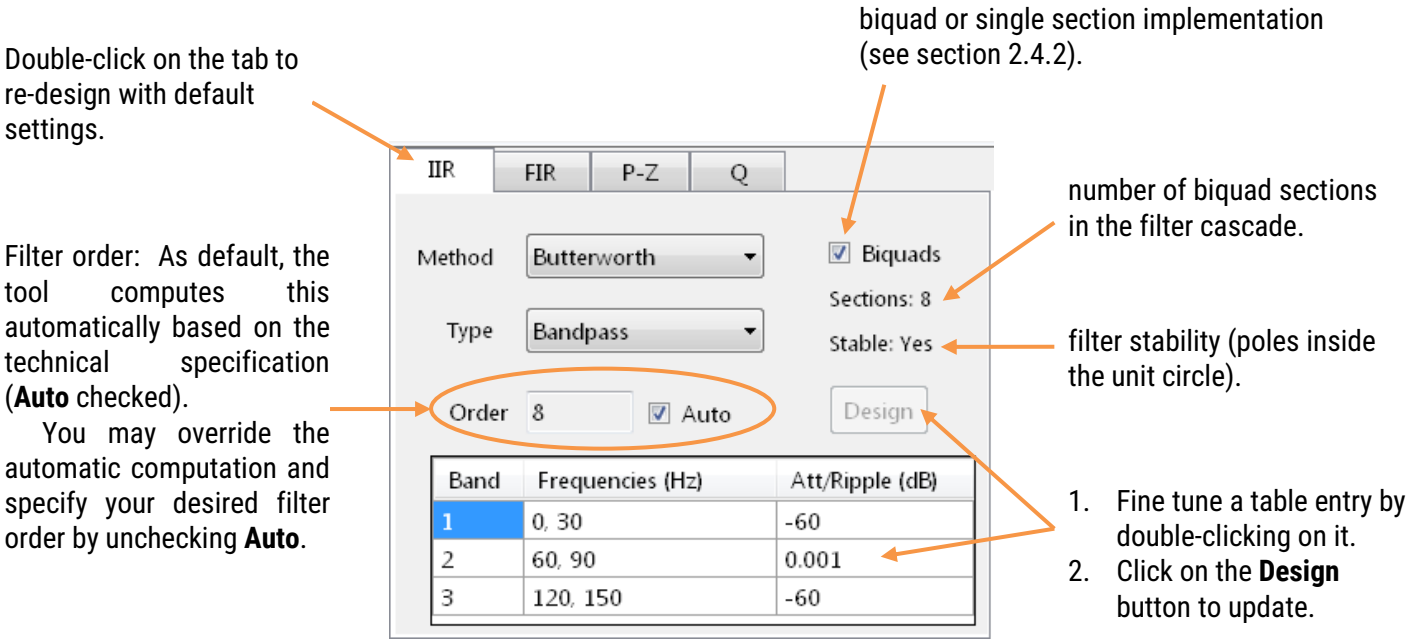


- Equiripple in both the passband and stopband.
- Fastest roll-off.
- Lowest order filter of all supported prototypes.

The frequency response charts shown pre-leaf show the differences between the various design prototype methods for a 5th order lowpass filter with the same specifications. A seen, the Butterworth response is the slowest to roll-off and the Elliptic the fastest.

The Bessel prototype is not supported, as the Bilinear transform warps the linear phase characteristics.

2.4.1. IIR Designer GUI



Filter orders of up to `100` (professional version only) may be constructed. However, in the case of bandpass and bandstop design, the filter order, `N` becomes `2N`.

IIR designs may be extended upon by utilising the P-Z editor, by allowing designers to modify or create a new filter by editing, adding or deleting any poles or zeros. In this mode (**User defined**), the **Method** dropdown list changes to **User defined**, as the design is no longer categorised by an analogue prototype. The comprehensive editor options allow for the design and customisation of any combination of poles and zeros, including the re-optimisation of the filter structure for implementation - see section 8 for more details.

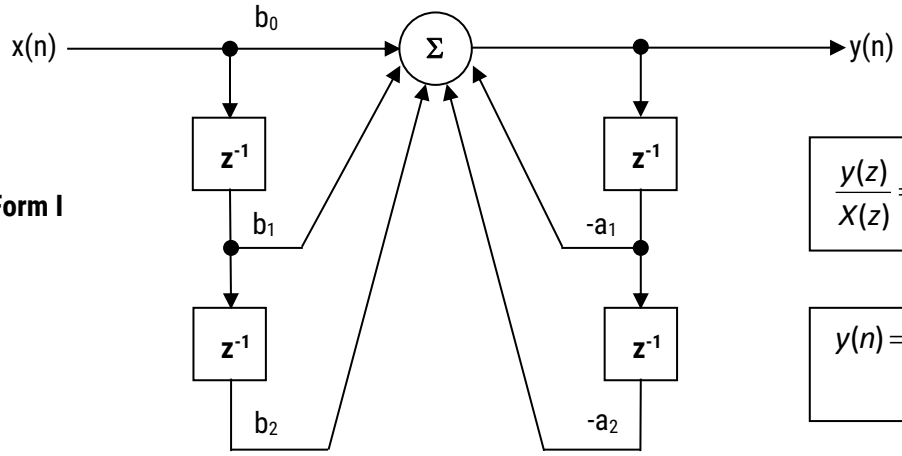
2.4.2. Biquads

All classical IIR filters are implemented as biquad filters (i.e. two poles and two zeros) as default. For any users requiring a single section implementation, simply uncheck the **Biquads** checkbox. However, as mentioned in section 2.3.8, higher filter orders generally lead to stability problems when poles are near to the unit circle.

The biquad implementation is particularly useful for fixed point implementations, as the effects of quantization and numerical stability are minimized. However, the overall success of any biquad implementation is dependent upon the available number precision, which must be sufficient enough in order to ensure that the quantized poles are always inside the unit circle.

The ASN filter designer supports the following three IIR filter structures:

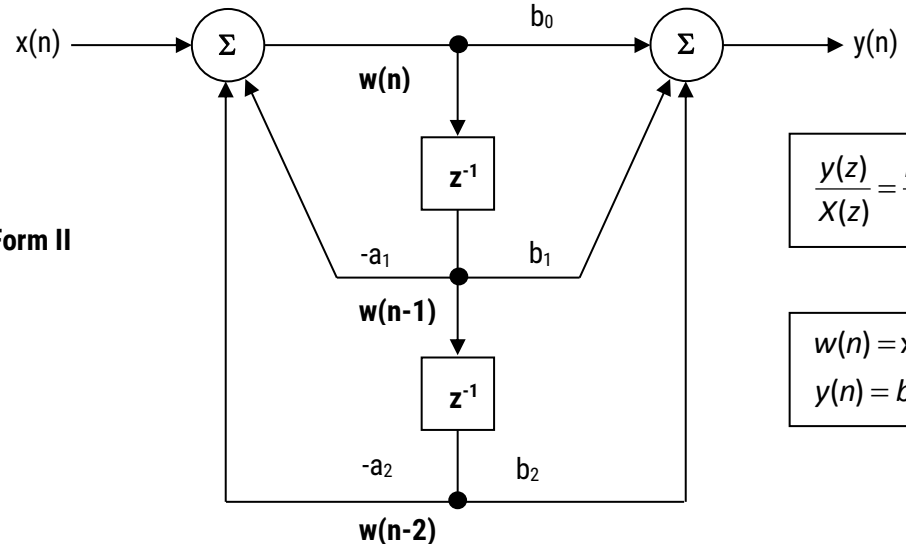
Direct Form I



$$\frac{y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

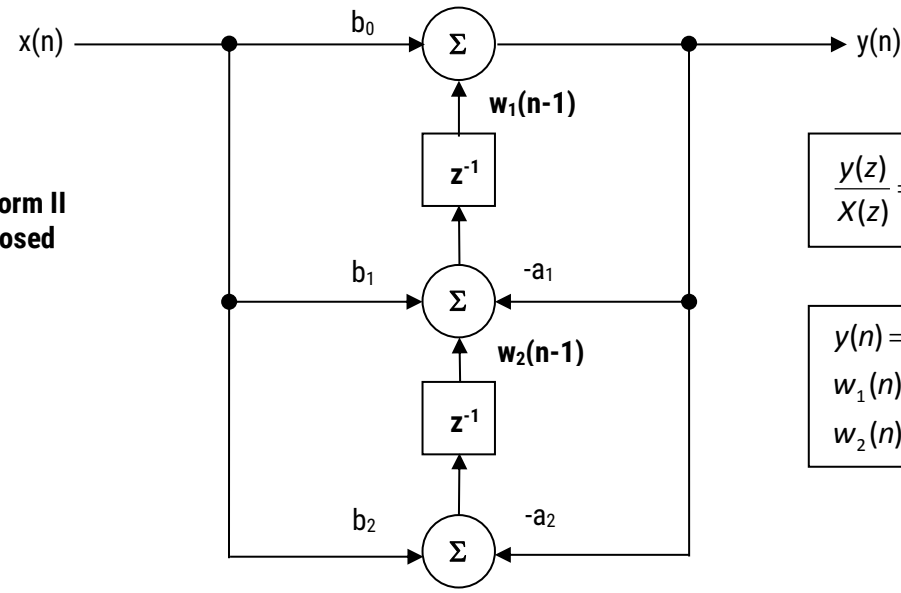
Direct Form II



$$\frac{y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$w(n) = x(n) - a_1 w(n-1) - a_2 w(n-2)$$
$$y(n) = b_0 w(n) + b_1 w(n-1) + b_2 w(n-2)$$

Direct Form II Transposed



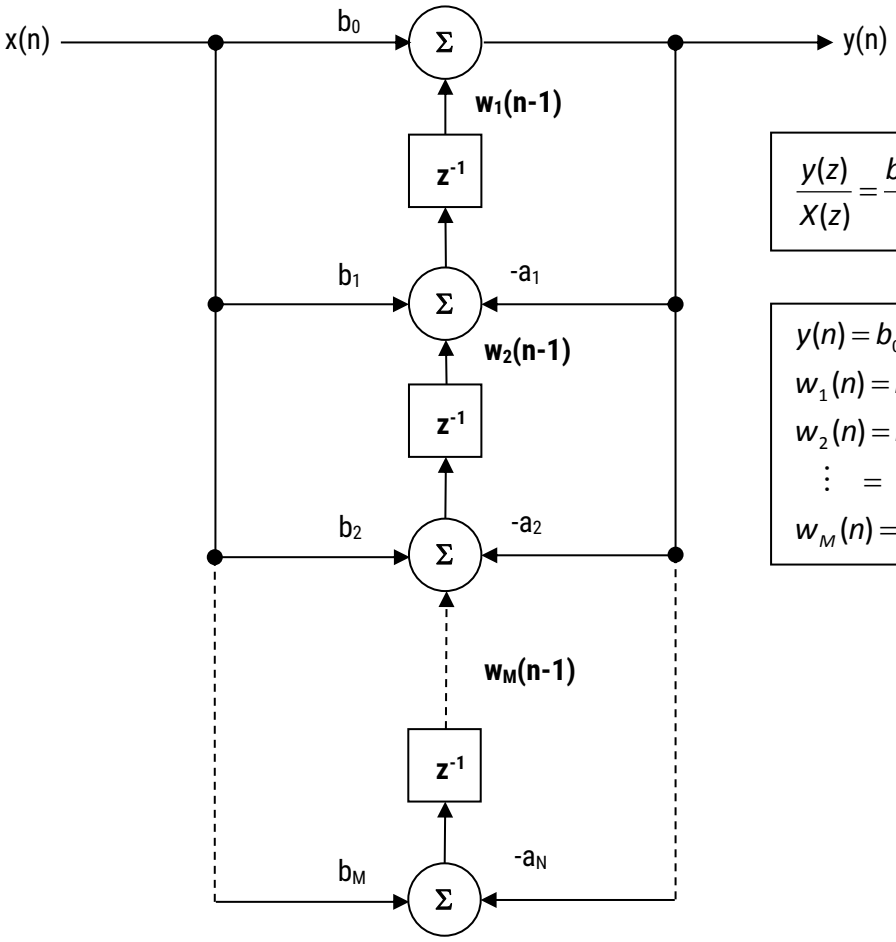
$$\frac{y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$y(n) = b_0 x(n) + w_1(n-1)$$
$$w_1(n) = b_1 x(n) - a_1 y(n) + w_2(n-1)$$
$$w_2(n) = b_2 x(n) - a_2 y(n)$$

Analysing the biquad structures, it can be seen that although the transfer functions are identical, the difference equations (i.e. implementation) are quite different. The **Direct Form II Transposed** structure is considered the most numerically accurate for floating point implementation, and is therefore the default filter structure. However, the **Direct Form I** is advocated for fixed point implementation by virtue of the single accumulator.

2.4.3. Single section IIRs

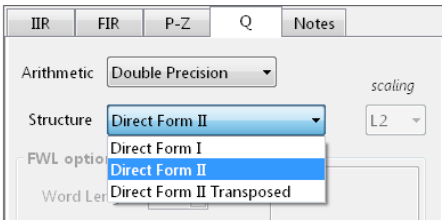
The ASN filter designer supports the design and implementation of both biquad and single section IIR filters. The concept of a **Direct Form II Transposed** single section filter is shown below for the case when *M=N*:



$$\frac{y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_N z^{-N}}$$

$$\begin{aligned} y(n) &= b_0 x(n) + w_1(n-1) \\ w_1(n) &= b_1 x(n) - a_1 y(n) + w_2(n-1) \\ w_2(n) &= b_2 x(n) - a_2 y(n) + w_3(n-1) \\ \vdots &= \quad \vdots \\ w_M(n) &= b_M x(n) - a_N y(n) \end{aligned}$$

2.4.4. IIR structure and Arithmetic options

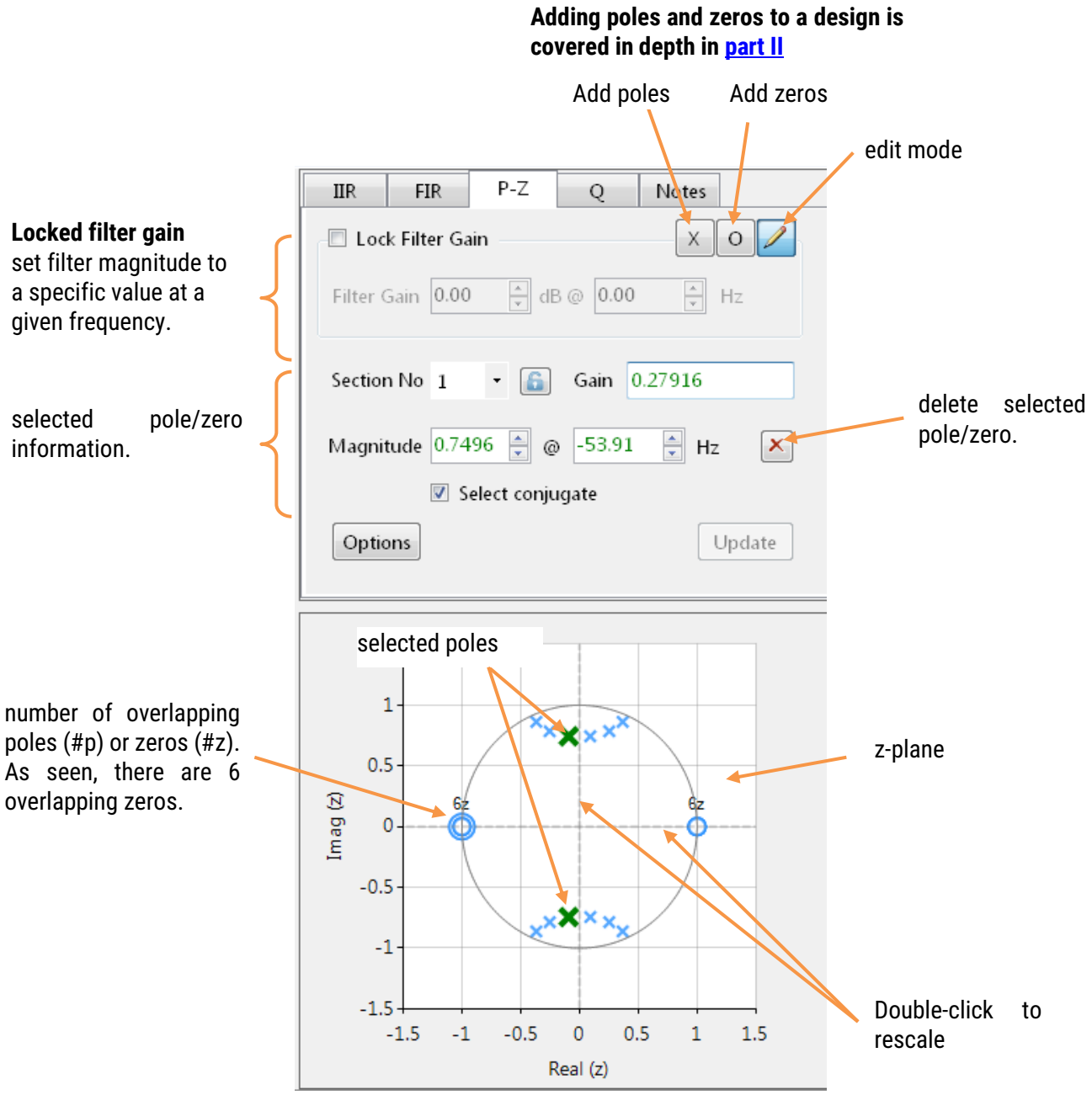


Filter structures and arithmetic options for both IIR and FIR filters may be found in under the **Q** tab.

NB. When using **Fixed point** arithmetic and the **Direct Form II** structure, the **scaling** option must be set - see section 5 for more details.

3. P-Z editor

The P-Z editor provides designers with a comprehensive but easy to use pole-zero editor, together with a few other useful options not commonly found in other filter design software.



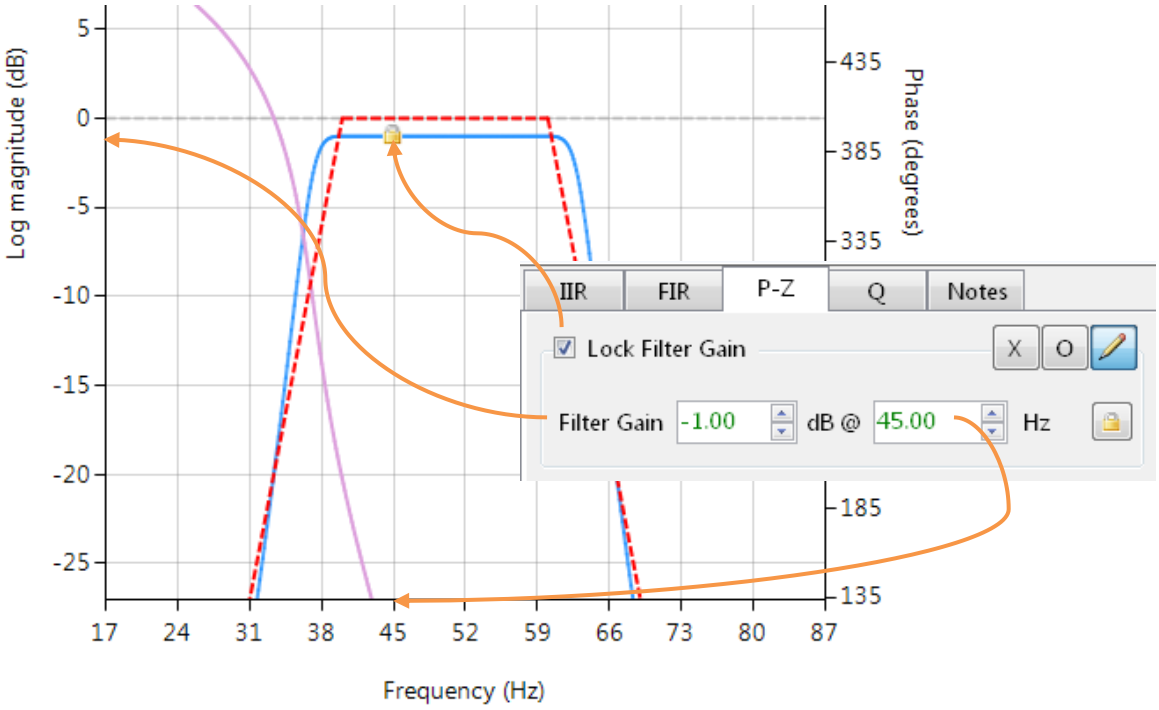
3.1. Zooming in/out and panning

As with the frequency response chart, you may zoom in and out into any area by using the mouse wheel.

Scrolling may be achieved by depressing the **left mouse button** and dragging the chart in any direction.

3.2. Locked filter gain

The **Locked Filter Gain** automatically calculates the filter gain required in order to set the magnitude spectrum to the specified gain in dB at the specified frequency. After clicking on the button, a yellow padlock will appear on the frequency response chart at the specified location.

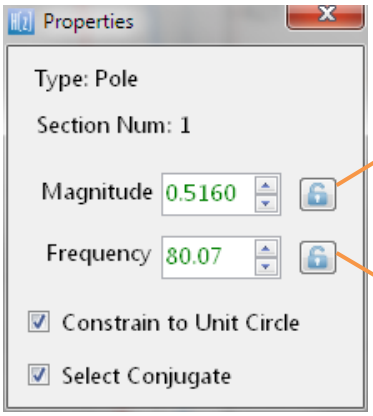


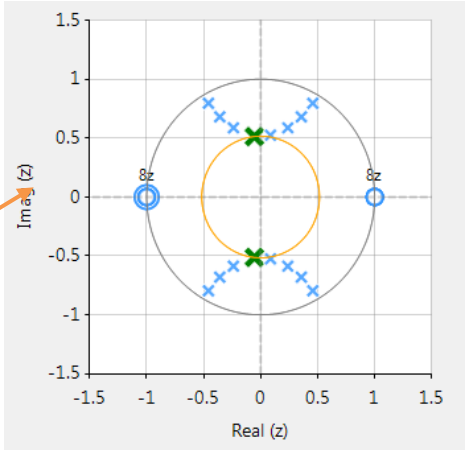
The specified **Filter Gain** is set to -1dB at 45Hz.

The exact **Locked Filter Gain** value will appear in the filter summary, and is automatically included into the filter implementation via the signal analyser.

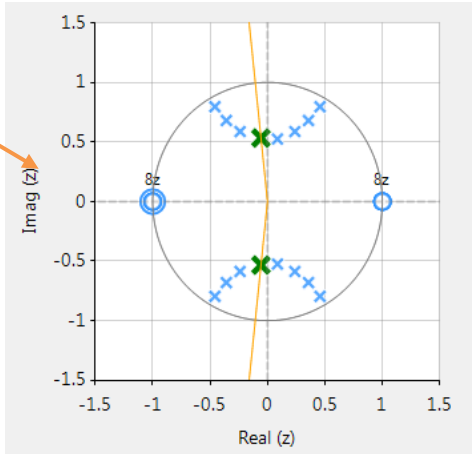
3.3. Pole-zero properties window

You may get the specific properties of a **pole** or **zero** by double-clicking on it in the P-Z chart.





Locked magnitude



Locked frequency

Locking the magnitude or frequency results in orange guide lines (see right) appearing on the chart. This is functionality is extremely useful for locking a dimension when modifying pole/zero position with the mouse.

3.4. Section number and section lock

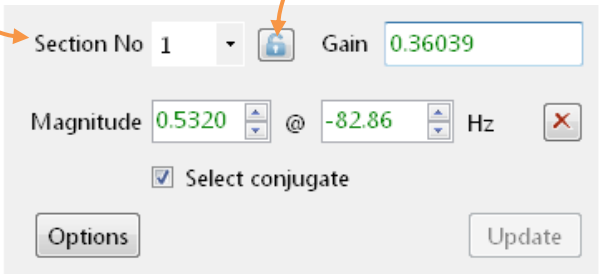
Section number

This allows you to highlight the pole and zeros of a specific section in the H1 filter:

- For FIR filters or single section IIR filters: the section number will always be equal to `1`.
- For biquad IIR filters: this will be a list of all the biquad sections in the filter cascade.

Section lock

Clicking on the section lock, allows you to focus on a specific section by highlighting all of the poles-zeros of the selected section number (**Section No**) and minimising the rest.



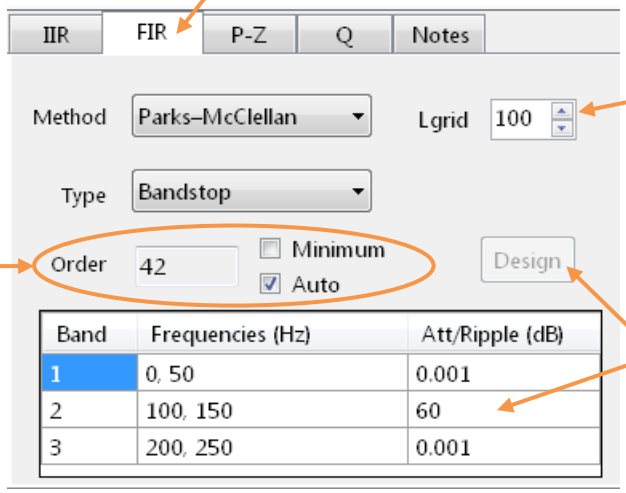
3.5. FIR Filter design

The FIR (finite impulse response) filter designer is implemented via the Parks-McClellan algorithm, and allows for the design of the following filter types:

- Lowpass
- Highpass
- Bandpass
- Bandstop
- Multiband
- Hilbert transformer
- Differentiator

The Parks-McClellan algorithm offers a degree of flexibility over other FIR design methods, in that each band may be individually customised in order to suit the designer's requirements.

Double-click on the tab to re-design with default settings.



Filter order: By default, the tool computes this automatically based on the technical specification (**Auto** checked).

You may override the automatic computation and specify your desired filter order by unchecking **Auto**.

Parks-McClellan algorithm grid step size parameter (16–900).

Fine tune a table entry by double-clicking on it.
Click on the **design** button to update.

Filter orders of up to 499 (professional version only) may be constructed, where this is limited to 200 for streaming audio applications. As with the IIR filters, an FIR's zeros may be modified by the P-Z editor (**Method** dropdown list changes to **User defined**), including the ability of adding poles and converting it into an IIR filter - see section 8 for more details.

Higher order FIR designs (>100): In order to speed up plotting performance, updates to the P-Z chart are postponed until the **left mouse button** is released.

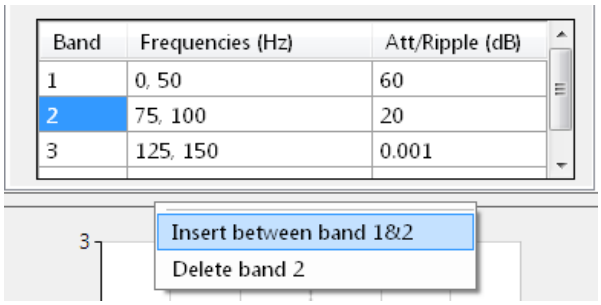
The order estimation of the Parks-McClellan algorithm may sometimes underestimate the filter order required for the given specifications. Therefore, in order to automatically increases the order estimate by 2 (overestimation) you may uncheck the **Minimum** checkbox.

3.5.1. Convergence and errors

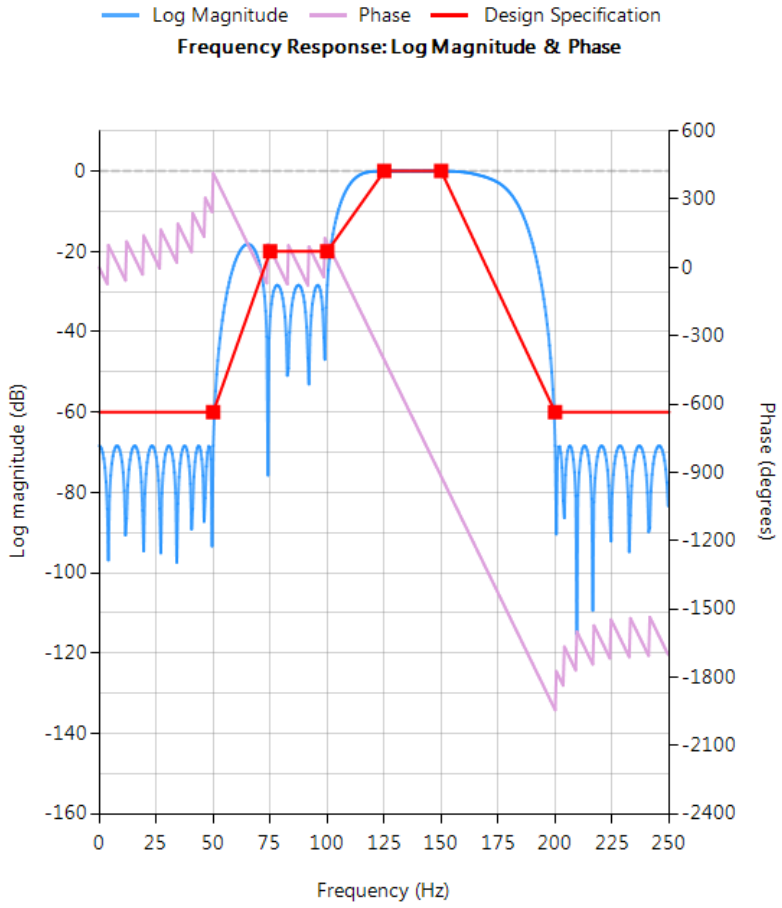
The Parks-McClellan algorithm is an optimal Chebyshev FIR design method, however the algorithm may not converge for some specifications. In such cases, increasing the distance between the design marker bands generally helps.

Errors in the root finding algorithm usually lead to undesirable results for high order filter implementations. As a consequence, the zeros presented in the P-Z chart for higher orders (> 60 or so) should only be interpreted as an illustration of the true positions. Also, if you are designing a high order FIR filter with a few hundred taps, it is not recommended to use the P-Z editor for editing the positions of the zeros.

3.5.2. Multiband FIR



In order to implement an arbitrary frequency response, you may use the **Multiband** design method. Extra bands may be added or removed from the design specification table by right-clicking on a **Band** and selecting the required option.



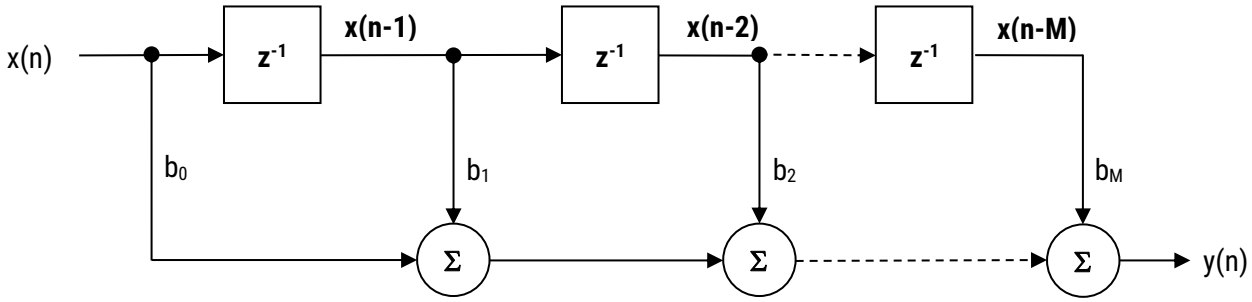
The design method requires that at least one band is a passband.

All bands with an attenuation of 10dB or less are classed as *passbands*. Depending on the level of band attenuation specified, the tool will automatically convert a stopband into a passband and vice versa.

3.5.3. FIR structures

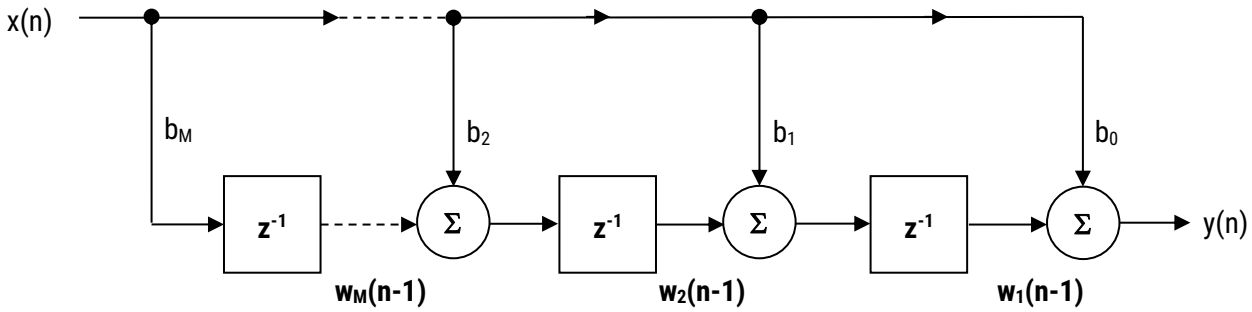
FIR (finite impulse response) filters are useful for a variety of signal processing applications, including audio signal processing and noise cancellation. Although several practical implementations for FIRs exist, the direct form structure and its transposed cousin are perhaps the most commonly used, and as such all designed filter coefficients are intended for implementation in a Direct form structure.

$$\frac{y(z)}{X(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_M z^{-M}$$



Direct Form structure

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \ldots + b_M x(n-M)$$



Direct Form Transposed structure

$$\begin{array}{lcll} y(n) & = & b_0 x(n) & + \; w_1(n-1) \\ w_1(n) & = & b_1 \mathrm{x(n)} & + \; w_2(n-1) \\ w_2(n) & = & b_2 \mathrm{x(n)} & + \; w_3(n-1) \\ \vdots & = & \vdots & + \; \vdots \\ w_M(n) & = & b_M \mathrm{x(n)} & \end{array}$$

The ASN filter designer supports the design and implementation of both **Direct Form** and **Direct Form Transposed** FIRs. As with IIR filters, the default structure is the **Direct Form Transposed** structure by virtue of its superior numerical accuracy when using floating point.

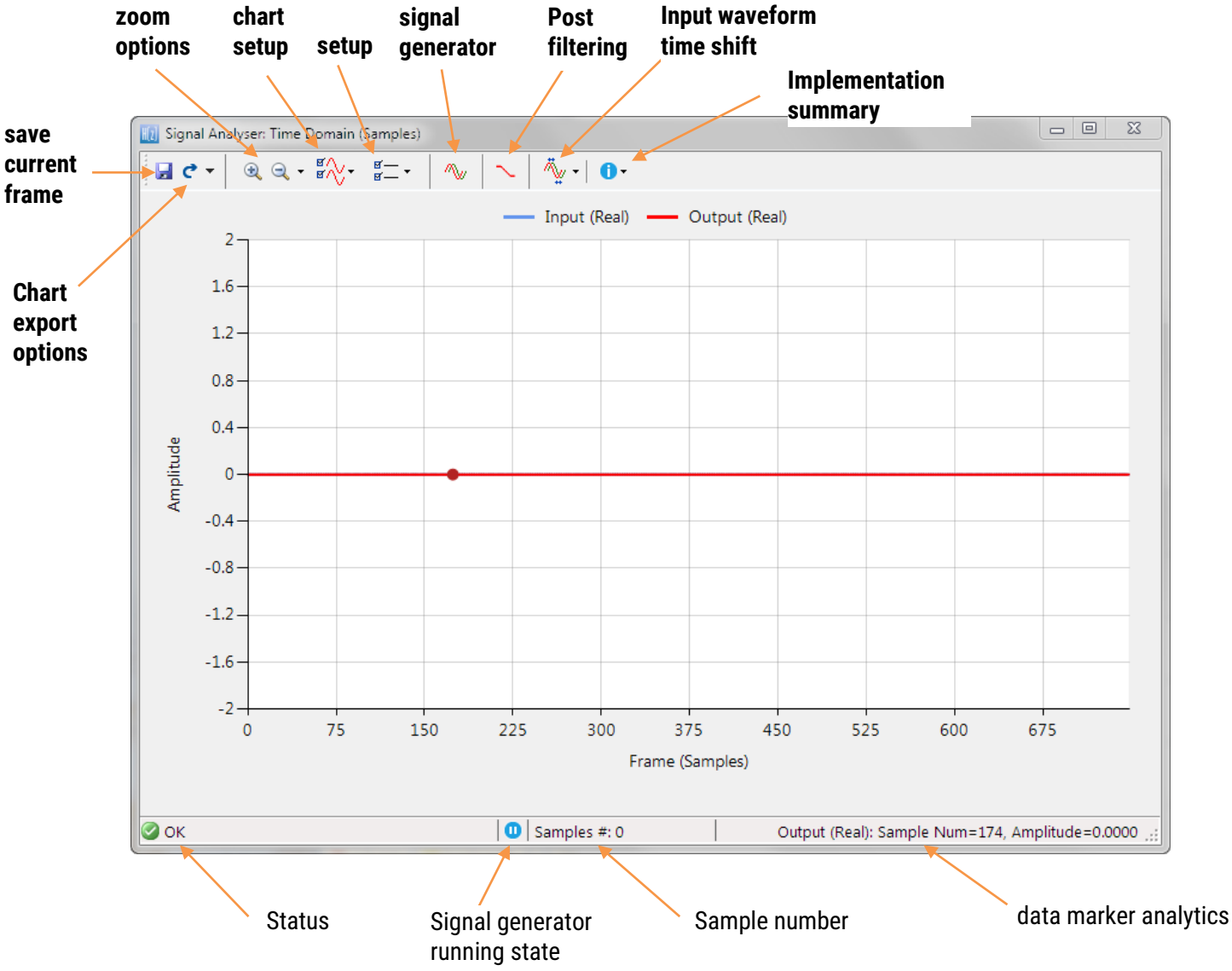
4. The signal analyser

You may start the signal analyser by clicking on the  button in the main tool bar.

The signal analyser allows designers to test their design on audio, real (user) data or synthetic data via the built-in signal generator. Default data playback is implemented as streaming data, providing a simple way of assessing the filter's dynamic performance, which is especially useful for fixed point implementations.

Both frequency domain and time domain charts are fully supported, allowing for design verification via transfer function estimation using the cross and power spectral density functions. As with all other charts, the signal analyser chart fully supports advanced zooming and panning, as well as comprehensive chart data file export options.

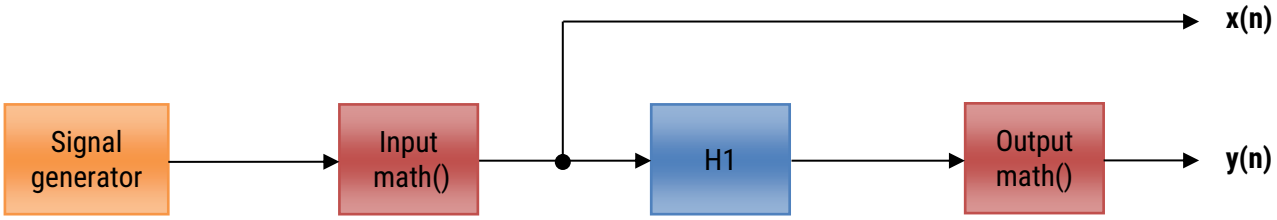
The signal analyser GUI is shown below.



4.1. Architecture

The signal analyser GUI is comprised of a time/frequency domain analyser and a signal generator. The GUI allows designers to explore the time and frequency characteristics of their designed H1 filter for various types of quantisation and inputs, but is flexible enough to also support analysis of 3rd party datasets. The signal analyser supports implementation for both real and complex coefficient filters, allowing for experimentation of the most demanding filter designs!

A block diagram of the signal analyser's architecture is shown below:



Sine, Square, Step, Sawtooth, White noise, External data file, Audio file etc.

FIR/IIR filter designed via the filter designer

As seen, the H1 filter is preceded and proceeded by optional **math()** function blocks which are useful for variety of signal processing operations, and may be independently enabled or disabled. If no mathematical function is required (Function() = **None**) the block is disabled and the data fed directly through. All operations are performed on complex data ($x = a + bi$), where the signal generator automatically converts real data into complex data by $x = a + 0i$. The following options are supported:

Function ()	Math operation	Description
None	-	Disable the function block.
Abs	$\|x\| = \sqrt{a^2 + b^2}$	Absolute.
Ln	$log_e\, x$	Natural logarithm.
Angle	$tan^{-1}\left(\frac{b}{a}\right)$	Compute the arctangent (phase in radians).
RMS	$\frac{\sqrt{a^2 + b^2}}{\sqrt{2}}$	Root mean square.
Sqr	x^2	Square.
Sqrt	$\sqrt{x}$	Square root.

In order to assess real-time performance of the filters, data from the signal generator is streamed (per sample) by default. However, in order to allow for impulse, step response and external data set evaluation, a blocked based mode is also provided - see section 4.2.1 for more information.

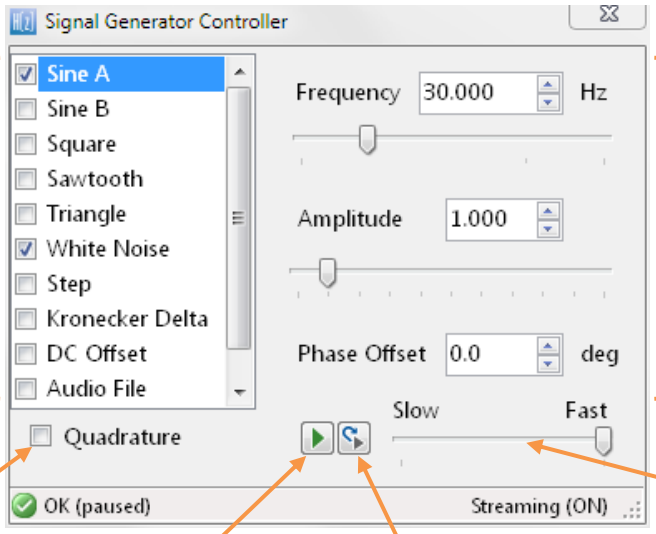
4.2. The signal generator

The signal generator allows you to test your filter with a variety of input signals, such as sine waves, square waves, white noise or even your own external test data.

The signal generator may be started by clicking on the button in the tool bar.

Signal methods
Enable the methods you wish to use by setting the checkmark. Click on the method in order to set its properties.

As seen on the right, the output signal is comprised of a 30Hz sinewave and white noise.



signal method properties

convert the signal method into a quadrature signal (real component in-phase and imaginary component shifted by 90 degrees).

playback speed (streaming only). Adjust the chart update speed.

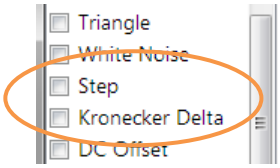
NB. This is disabled when streaming audio.

play/pause signal generator.

Reset signal generator and re-run.

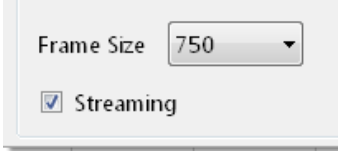
Two independent sinewaves (**Sine A** and **Sine B**) are available, allowing you to experiment with simple signal configurations for a variety of practical applications.

4.2.1. Impulse, step response and short external data set evaluation



You may evaluate your filter's impulse and step response characteristics by enabling either the **Kronecker Delta** method (impulse response evaluation) or the **Step** method (for step response evaluation). By default, the amplitudes are set to `1.000`, but may be changed as required.

Finally, for instant results, block based mode should be selected by unchecking the **Streaming** checkbox in the Setup menu. This feature is covered in depth in the following video tutorial.

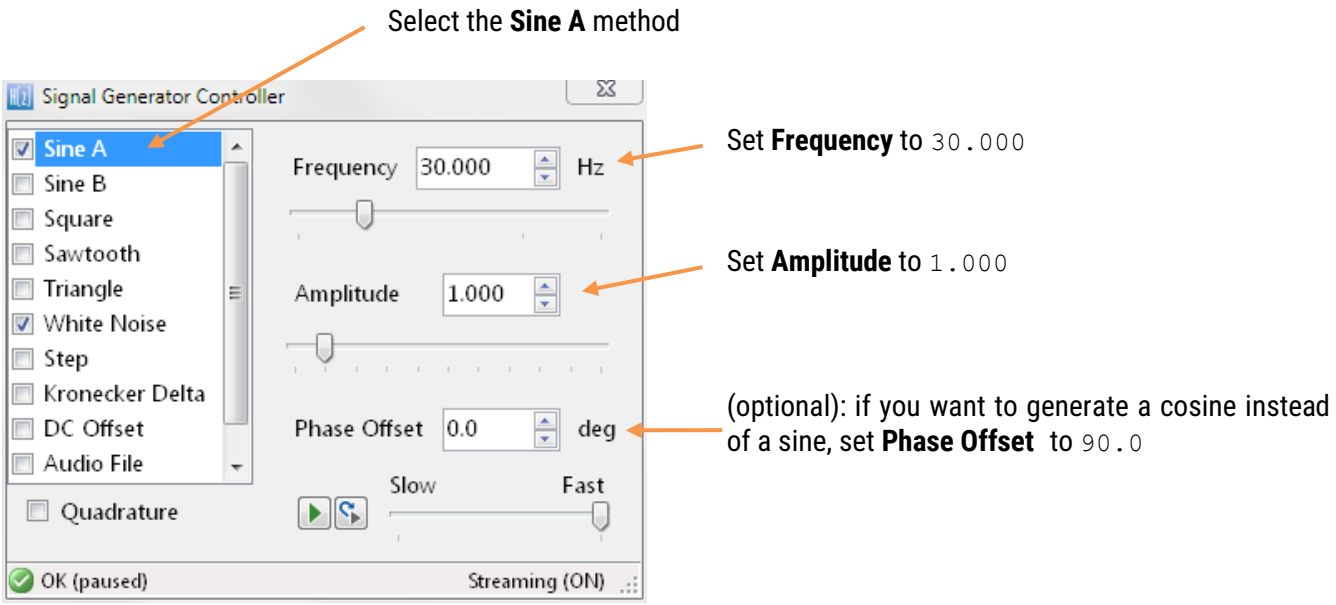


Block base mode is an extremely useful method of evaluation of short external data sets (see section 4.3.7 on how to load them), as the filtering performance on datasets less than or equal to the selected **Frame Size** can be instantly visualised and optimised accordingly.

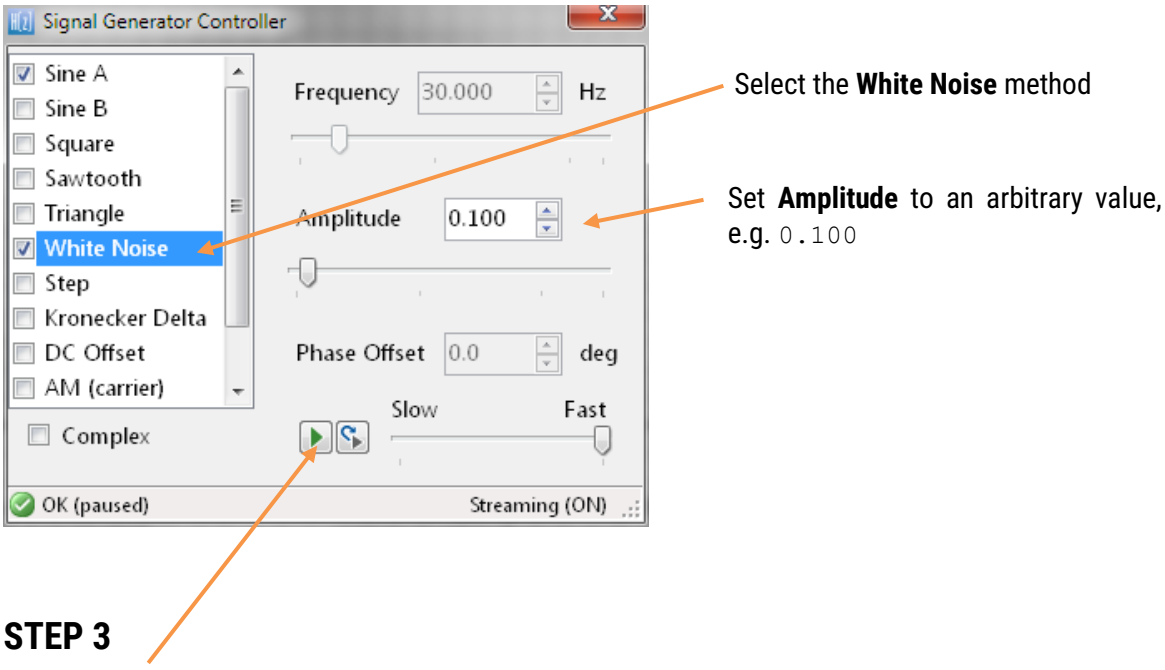
4.2.2. Streaming your first application with the tool

As an example application, let us assume that we want to test a designed filter with a 30Hz sinewave of amplitude `1.000` with some additive White noise. This can be simply achieved by setting the generator up as follows:

STEP 1



STEP 2



STEP 3

Play the signal generator.

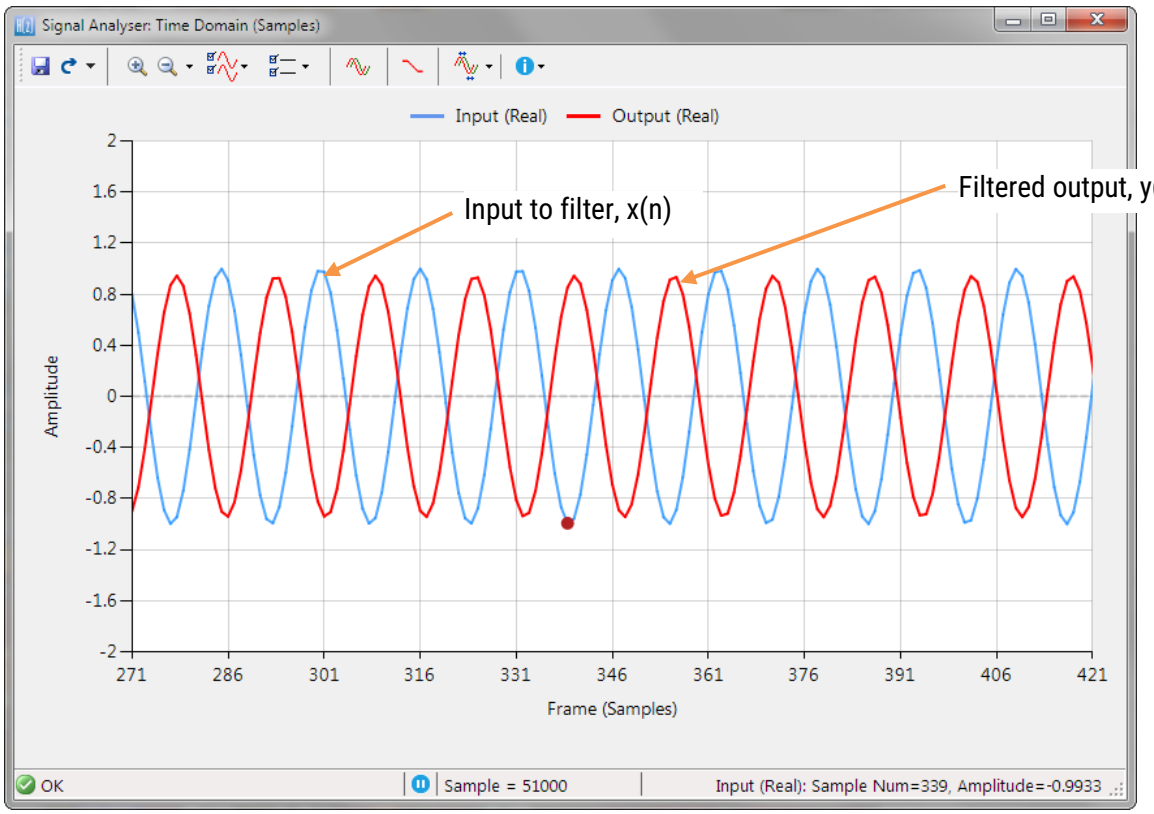
4.3. Data analysis

Data analysis is performed with the mouse. Where, moving the mouse over the chart will automatically produce data markers and data analytics (shown at the bottom right side of the GUI). The signal analyser is directly coupled to the filter designer GUI. This means that you may modify the filter characteristics and see the effects in real-time in the signal analyser. This functionality very useful when designing audio filters, as the new filter settings can be heard immediately on the streaming audio feed as discussed later on in section 4.3.6.

When conducting frequency analysis, the data analysis algorithm implements a specialised version of the Discrete Fourier transform, which allows designers to perform high resolution frequency analysis of any point of interest on the magnitude spectrum respectively.

4.3.1. Time domain analysis

Upon clicking the signal generator's play button the signal analyser window will be updated.



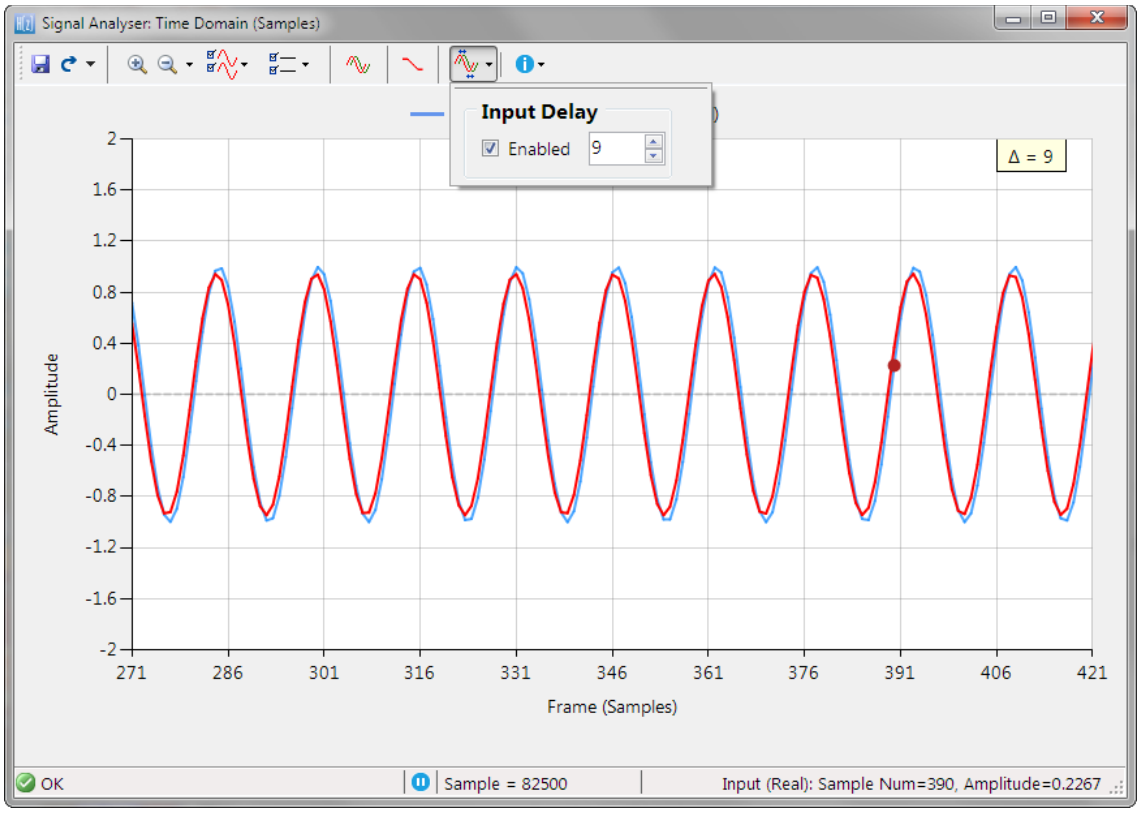
As seen, the signal analyser resembles an oscilloscope, where live data from the signal generator is fed (streamed) into the H1 filter on a sample by sample basis. You may perform data analysis on the chart data by panning, zooming with the mouse.

Signal generator playback speed (streaming only): You may adjust the chart update speed by setting the slider accordingly.

NB. This is disabled when streaming audio.

4.3.1.1. Delaying the input waveform

The tool offers designers the ability to delay the input waveform in order to visually compensate for the effects of phase/group delay.



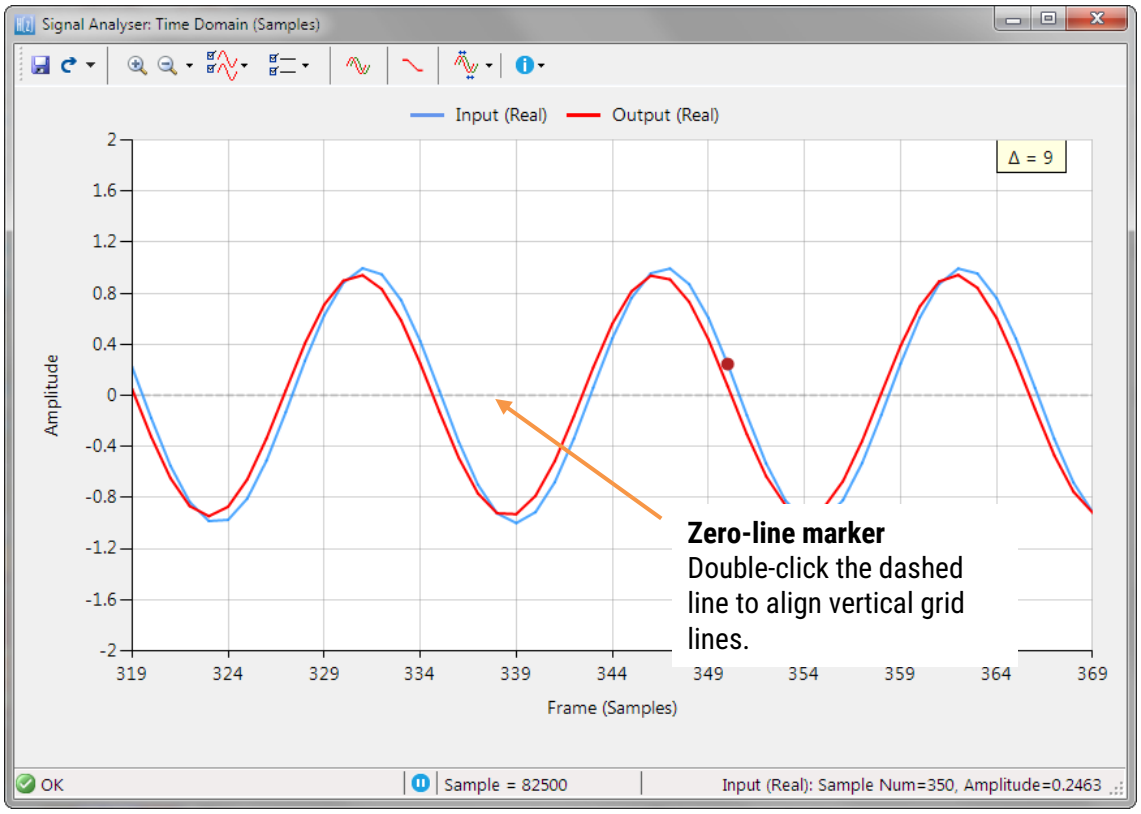
This feature does not affect the original waveform, and is only intended for display/analysis purposes.

4.3.1.2. Input and Output waveform traces

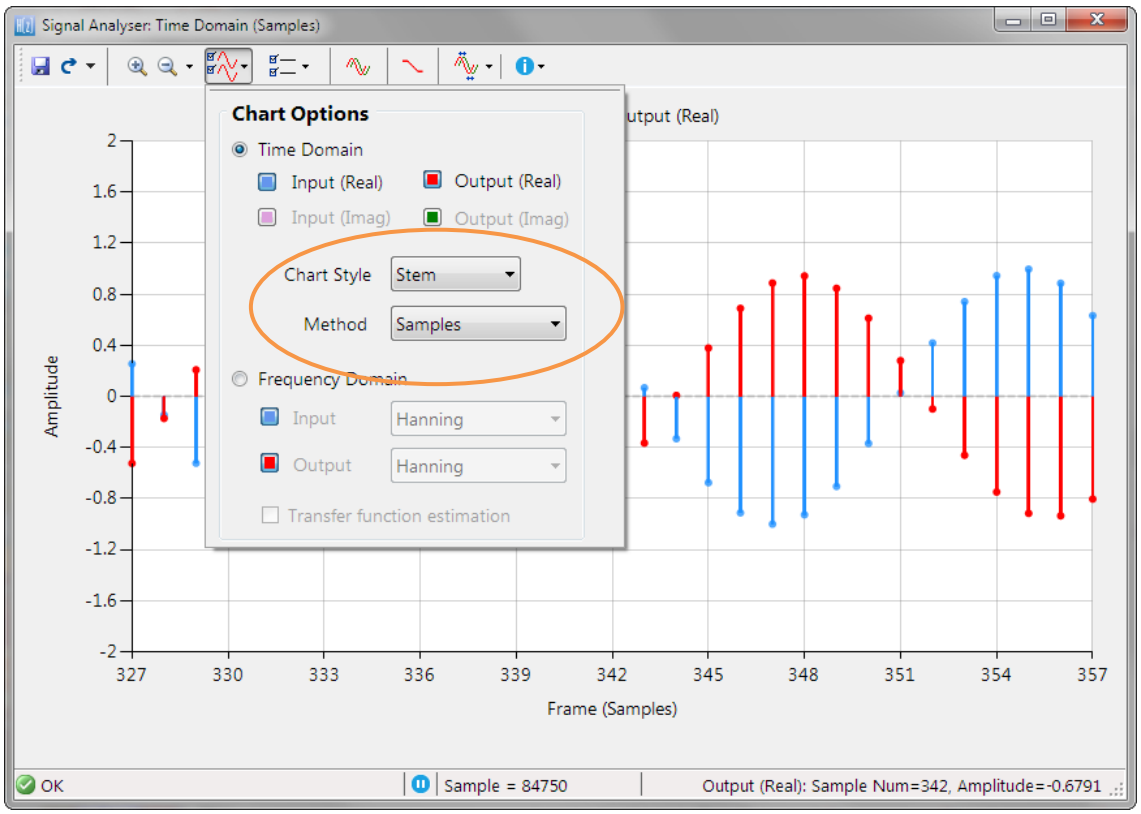
The signal generator automatically converts its output signal into a complex signal. The options **Real** and **Imag** refer to the real and imaginary components of the input and output signal respectively.

As complex filters are somewhat of a specialty, the default display setting is **Real** for both input and output signals.

4.3.2. The zero-line marker

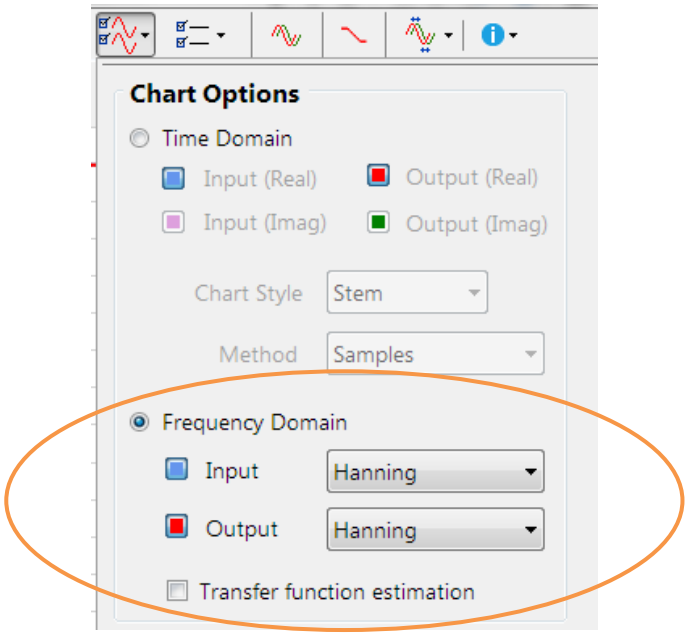


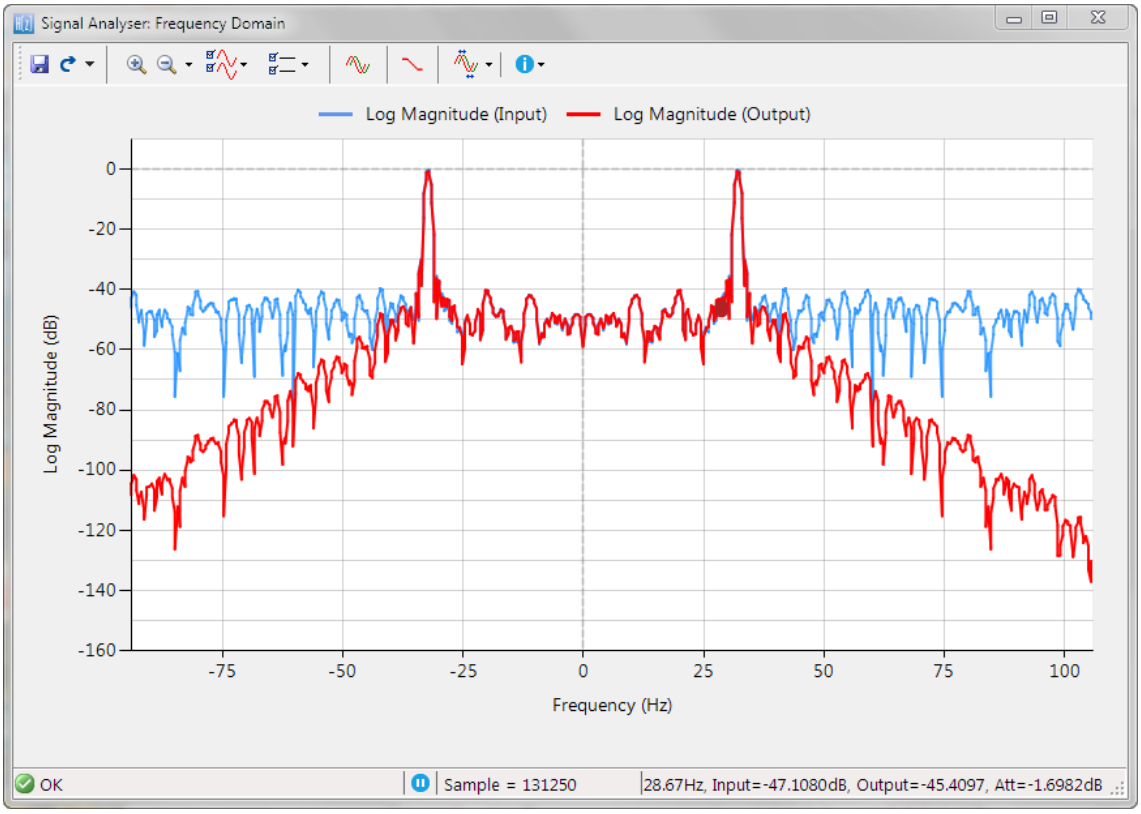
4.3.3. Stem chart



4.3.4. Frequency domain analysis

The signal generator's default view is time domain analysis, but you may also perform frequency domain analysis by altering the chart options, as shown below:





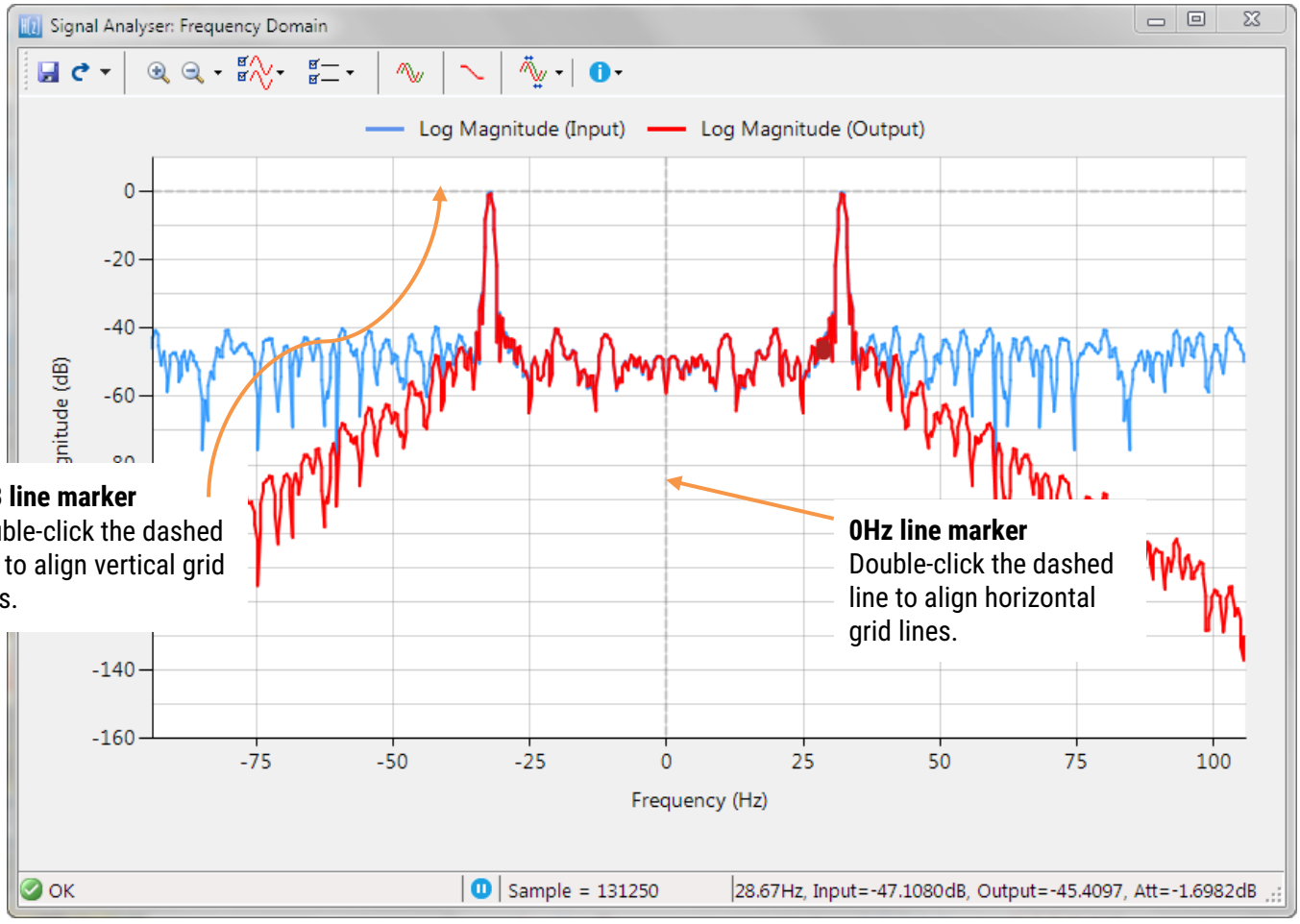
As with the time domain chart, you may perform data analysis on the frequency domain chart data by panning, zooming with the mouse.

4.3.4.1. Input and Output waveform traces

Two traces are used for the frequency domain analysis (as seen on the left). Although when **Transfer function estimation** is enabled, only the **Output** (red trace) is shown.

The colours of the traces cannot be altered!

4.3.5. The 0dB and 0Hz line markers

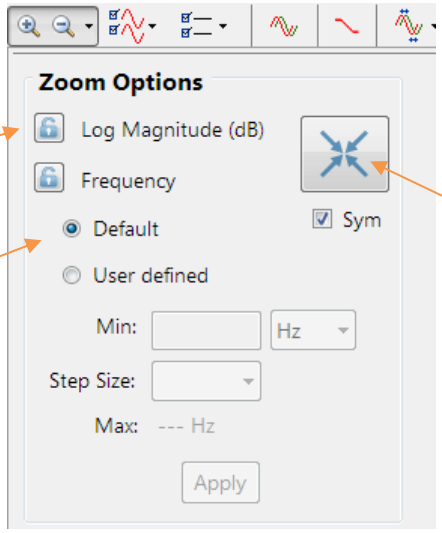


0dB line marker
Double-click the dashed line to align vertical grid lines.

0Hz line marker
Double-click the dashed line to align horizontal grid lines.

4.3.5.1. Chart zoom options

The signal analyser provides designers with a comprehensive zooming menu for undertaking analysis of demanding signals.



Lock/unlock magnitude axis

Default zoom
allows zoom with the mouse wheel over the range ±Nyquist.

reset zoom on all axes
If **Sym** is checked, then the frequency axis is reset to ±Nyquist, otherwise the reset is between 0-Nyquist.

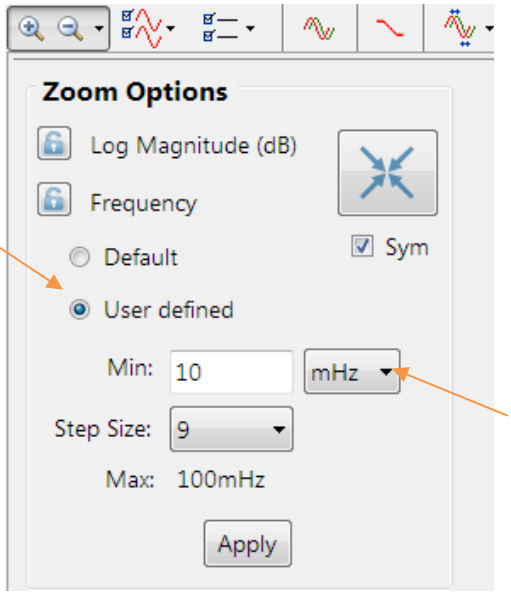
4.3.5.1.1. Locking axes

In order to simplify data analysis, you may lock a specified axis for zooming/panning purposes. This has the advantage of allowing you to customise each chart axis to your exact requirements.

4.3.5.1.2. Zooming to a specific frequency range

user defined zoom
You may zoom to a specific frequency range with the **User defined** zoom function.

The universality of this function allows you zoom to **mHz** resolution even when the sampling rate is in the **MHz** region!



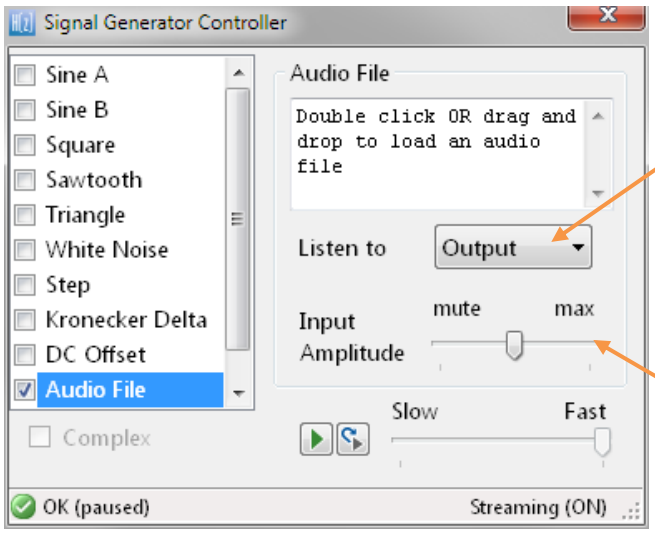
Choose the frequency scale that you wish to zoom to. Notice here that we are setting the x-axis to the range: `10-100mHz`.

This functionality is covered in the following video tutorial

Panning is disabled on the x-axis (frequency) when this function is enabled!

4.3.6. Audio file

The signal generator allows you to load **.wav** audio files for playback via the **Audio File** method. Both mono and stereo formats are fully supported for the following sampling rates: `8.000`, `11.025`, `16.000`, `22.05` and `44.1kHz`. There is no restriction as to the length of the **.wav** file.



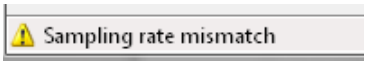
choose what you want to listen to (i.e. the input or output signal)

Adjust the amplitude of the input signal.

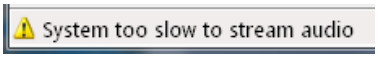
You may add extra signals to input audio stream by enabling the methods as discussed above. If using the professional version, a maximum limit is placed on the filter order that can be evaluated. For an IIR filter this is set at `20`, and for an FIR this is set at `200`.

All audio input signals are normalised and converted into floating point format for use with signal generator. Adjusting the input amplitude to `>±1` will result in signal distortion.

If the sampling rate of the loaded audio file does not match the filter's sampling rate, you will receive a warning message. You may still continue with your experimentation, but with the understanding that the audio stream and filter are mismatched.



For higher sampling rates, such as 44.1kHz the UI may become sluggish on some computers. Internal analytics monitor the responsiveness of the UI and if deemed too sluggish, audio playback will be paused with a warning message in the signal analyser UI.



For converting other audio formats, such as mp3 and ogg into .wav the reader is referred to the freely available open source audio editing program Audacity (http://web.audacityteam.org/)

4.3.7. External data file

The signal generator allows you to load external data files for playback via the **Data File** method. Two types of file format are supported: CSV (comma separated value), and single column data as shown below.

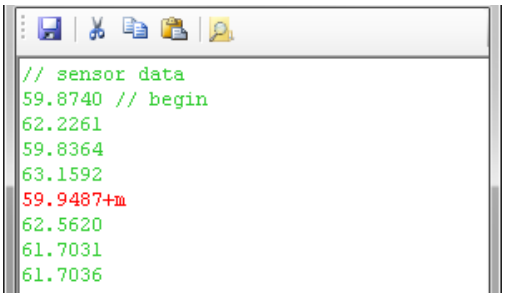
Data must be a single column text file and may contain real or complex values (**i** or **j**) and user comments (**//**).

Example 1	Example 2	Example 3

`// sensor data`
`59.8740`
`62.2261`
`59.8364`
`63.1592`
`59.9487`
`62.5620`
	`// sensor data`
	`59.8740+10j`
	`62.2261`
	`59.8364`
	`63.1592`
	`59.9487+i`
	`// marker data`
	`62.5620`
		`// sensor data`
		`59.8740+10i //begin`
		`62.2261`
		`59.8364`
		`63.1592`
		`59.9487-2.4i`
		`62.5620`

A maximum data length of `20,000` values may be loaded within a min-max data range of `±9000`.

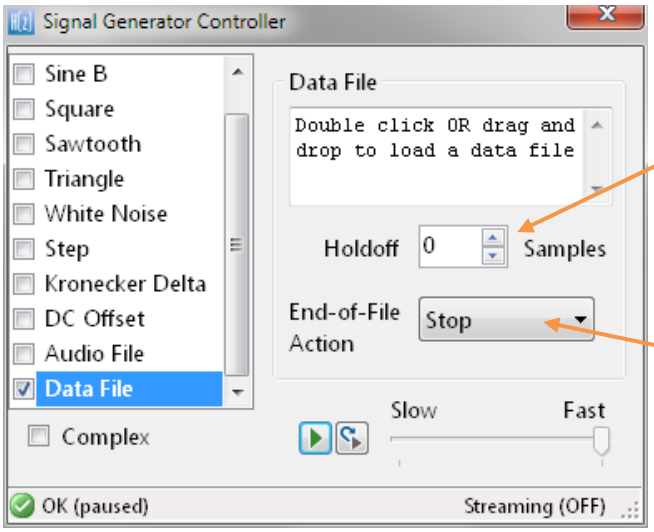
4.3.7.1. Correcting errors



If the import engine detects any errors (single column data file only), a file viewer window is opened and the error(s) highlighted in red (see left).

Use the toolbar options to edit the file, and then click on to  re-save the file. After re-saving you need to import the file again.

4.3.7.2. Menu options



offset the data set by the specified number for samples: Use this option in conjunction with the **White Noise** method in order to pre-fill the filter's data lines with non-zero data.

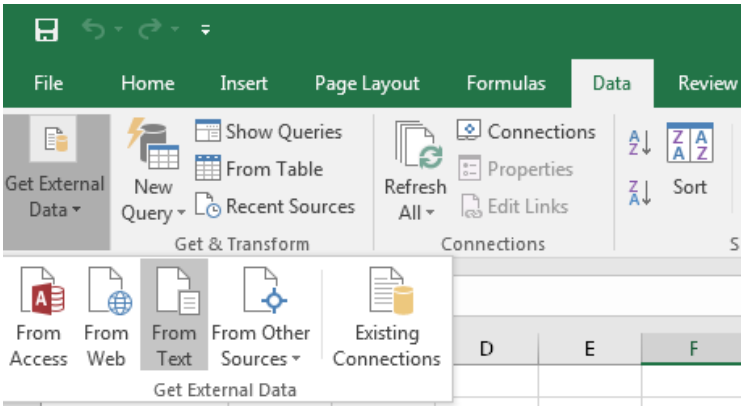
choose what action should be performed at the end-of-file:

- **Stop** playback
- **Repeat** from the beginning
- **Continue** streaming (if enabled)

4.3.7.3. Importing data from a Microsoft Excel spreadsheet

Many sensor datasets are available in Microsoft Excel spreadsheets. In order to export a spreadsheet's dataset to the ASN filter designer for analysis, the following steps should be followed:

1. Import the datafile and select the column of data that you wish to import into the filter designer:

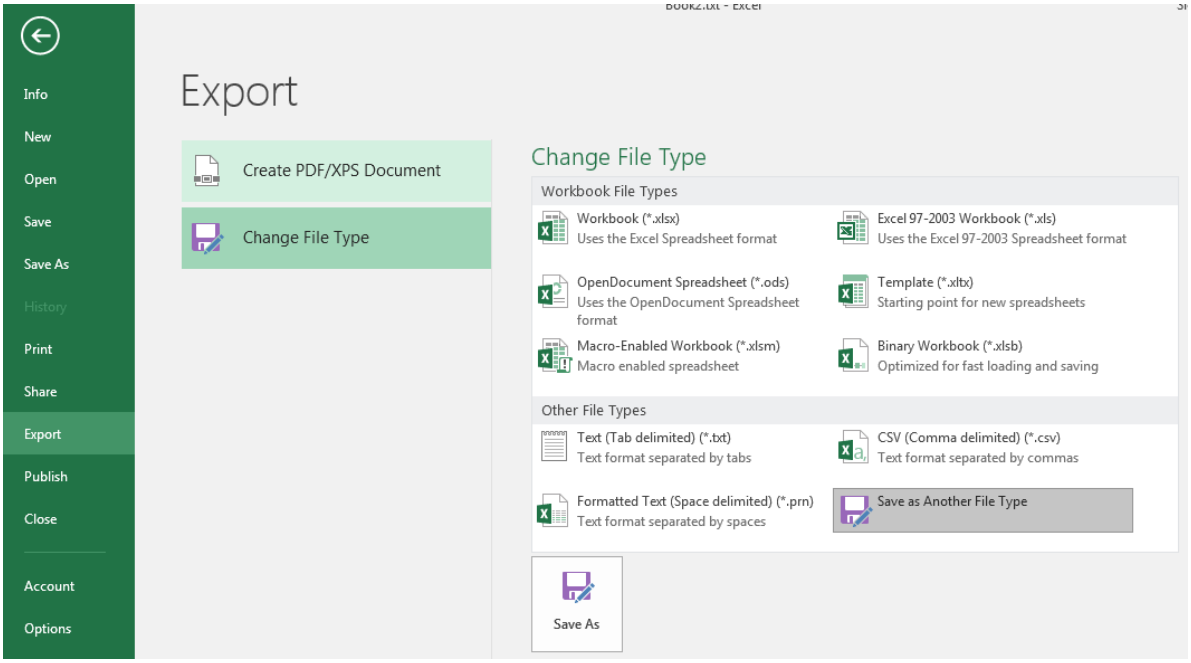


	A	B	C
1	sensor A	sensor B	sensor C
2	3383.352	8170.104	-11553.5
3	2479.477	8821.531	-11301
4	1560.315	9418.571	-10978.9
5	631.533	9957.542	-10589.1
6	-301.142	10435.12	-10134
7	-1231.96	10848.37	-9616.41
8	-2155.18	11194.73	-9039.54
9	-3065.12	11472.07	-8406.95
10	-3956.16	11678.68	-7722.52
11	-4822.81	11813.29	-6990.48
12	-5659.72	11875.06	-6215.34
13	-6461.74	11863.62	-5401.89
14	-7223.92	11779.04	-4555.13

2. Paste the dataset into a new workbook.

3. Save the new workbook as a Text file:

Export > Save as Another File Type > Text (MS-DOS) *.txt



4.3.8. Amplitude modulation

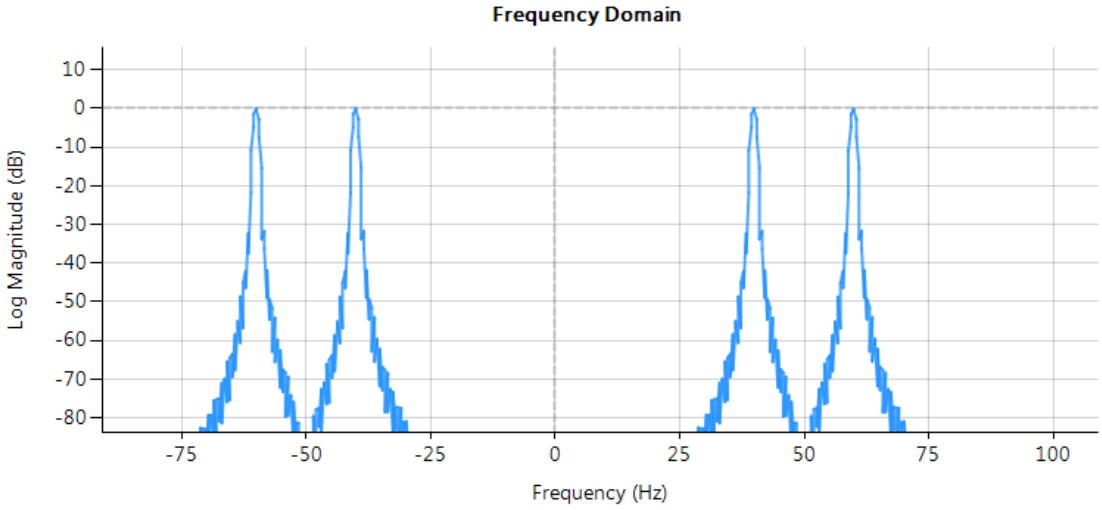
The signal analyser fully supports AM [amplitude modulation, **AM (carrier)**]. The type of AM implemented is the so called double sideband-suppressed carrier modulation given by:

$$y(n) = A_c \cos(2\pi f_c n/f_s)\, m(n)$$

where, $m(n)$ is the summed output of the other enabled signal generator output signals (e.g. **Sine A, Sine B, White Noise** etc.). The resulting output spectrum is given by:

$$Y(f) = \frac{1}{2} A_c [M(f - f_c) + M(f + f_c)]$$

The example shown below illustrates the resulting spectrum for a 10Hz sinusoid amplitude modulated with a 50Hz carrier signal ($A_c = 2$).



As expected, the resulting spectrum has two peaks (40Hz and 60Hz) centred around ±50Hz.

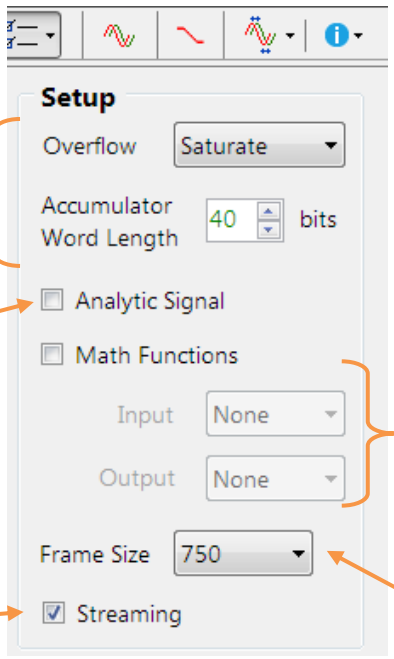
4.3.8.1. Practical application

AM has found particular use in the sensor world when performing accurate strain measurements using a loadcell sensor excited by an AC source. In such an application, the carrier frequency, f_c and an excitation sinusoid are the same frequency, and the phase offset (due to instrumentation electronics) between the two sinusoids is considered to be < 0.1 degree, which has minor impact on the estimate. Using the theory developed above, it can be seen that any unwanted DC offsets from a instrumentation amplifier and ADC also present in $m(n)$ will be moved to f_c and the desired sinusoid moved down to DC (0Hz). The amplitude of the sinusoid can now be easily extracted with a simple lowpass filter, which will smooth the output by filtering out the unwanted components higher up in the spectrum.

4.4. Setup

The Setup menu allows you to customise the fixed point quantisation settings, input/output mathematical functions, frame size and select between streaming and blocked based mode.

Fixed point quantisation settings
- set the overflow rules: **Saturate** or **Wrap**.
- set the accumulator length: `40 bits` is the default.



Hilbert transform only
Automatically creates an analytic signal.
See below for more information.

I/O math functions
In order to condition the input and output data, six extra mathematical functions are available:
Abs, **Angle**, **Ln**, **RMS**, **Sqr** and **Sqrt**

selects streaming or block based mode (see section 4.2.1 too).

Frame size
select analyser frame size, between `100-1500`.
NB. this setting will be automatically altered when streaming audio and computing the transfer function.

4.4.1. The Hilbert Transform and the analytic signal

Checking the **Analytic Signal** checkbox will automatically delay the input data stream by N/2 (where N is the filter order) and re-order the filter coefficients in order to produce an analytic signal. The delayed input signal (real component) is also pre-filtered with a first order Butterworth highpass filter in order remove any DC components. Where, the cut-off frequency point (-3dB) of the filter is equal to one-fifth of Band 1's upper frequency value respectively.

4.4.2. Block based and streaming mode

Block based mode will process a complete frame of input and output data and then reset the signal generator to its initial conditions. This functionality is extremely useful for instant evaluation of a sinewave's initial phase shift as well as instantly evaluating the filter's impulse and step response respectively. However, for many cases, the near instant update will seem like the GUI has frozen or is inactive, as the White noise generator seed (for example) will be reset to its default value.

Streaming is the default setting and is used to assess real-time performance of the filters, where data from the signal generator is streamed (per sample) indefinitely. You may set the playback (chart update) speed by adjusting the playback slider on the signal generator, as discussed in section 4.2.

You may reset the signal generator by clicking on the **re-run** button.

When streaming audio the GUI enters a special type of streaming mode, whereby the playback slider and frame size controls are disabled.

4.5. Chart options

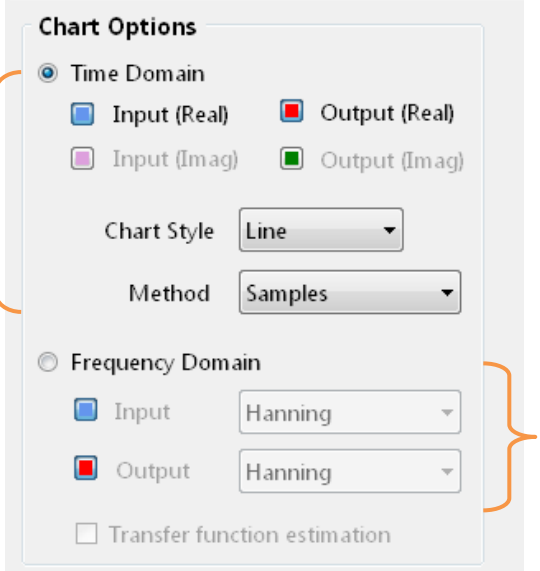
Chart options configures the chart for time or frequency domain analysis.

Time domain options
There are four chart traces (lines), representing the real and imaginary components of the Input and output signal respectively. You may enable or disable the traces that you wish to view.

Two lines styles are supported: **Stem** and **Line**.

Four analysis methods are available:

- Samples
- Phase
- Biased autocorrelation
- Real Cepstrum



Frequency domain options
There are two chart traces (lines), representing the log magnitude input and output spectra respectively. You may enable or disable the traces that you wish to view, and specify which smoothing window function is used.

4.5.1. Biased autocorrelation

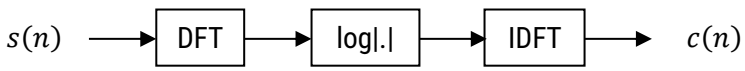
The biased auto-correlation is given by:

$$R_{xx}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n-k); k \geq 0$$

Autocorrelation is useful for finding periodic patterns, such as the period of sinewave buried in noise.

4.5.2. Real Cepstrum

The real Cepstrum is deconvolution technique heavily used in speech and audio applications. The essence of the Cepstral operation centres around use of the DFT (discrete Fourier transform) and a log operator in order to deconvolve the transfer function (i.e. the slowly varying component) from the excitation (the faster moving component). The Cepstral deconvolution process may be described by the following block diagram:



Mathematically, considering $s(n)$ to be a convolution of an unknown transfer function, $h(n)$ and an excitation $e(n)$, we may write (where, FFT and IFFT are computationally efficient methods for computing the DFT and inverse DFT):

$$s(n) = h(n) * e(n)$$

$$FFT\ [s(n)] = H(w).E(w)$$

$$\log|S(w)| = \log|H(w)| + \log\ |E(w)|$$

$$IFFT\ \{\log|S(w)|\} = c(n) = IFFT\ \{\log|H(w)| + \log\ |E(w)|\}$$

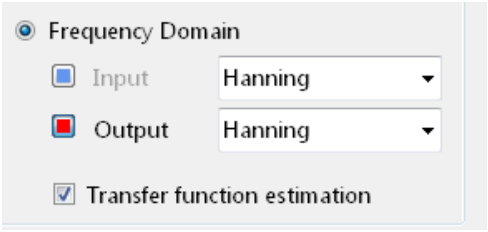
Entering the Cepstral domain, $c(n)$ notice how the transfer function and excitation are now *a linear combination* (i.e. additive) and such can be analysed separately. Notice also that unlike other pole-zero modelling methods, the Cepstrum may be used to model the effects of a system comprised of an unknown number of poles and zeros without any explicit knowledge of the system, as the analysis is non-parametric. However, care should be exercised when using this method for transfer function analysis, as no performance function is used, and as such the resulting coefficients are not strictly speaking optimal.

A new set of terminology was invented by the original author, and as such, 'frequency' was named 'quefrency' and 'spectrum' named 'Cepstrum'. The index of the Real Cepstrum (which is actually discrete time) is referred to as the quefrency axis.

4.5.3. Transfer function estimate

In order to validate the magnitude frequency response of the designed filter with sample data, the **Transfer function estimation** option is available. This method estimates the system transfer function by computing the quotient of the cross-power spectral density of **x** and **y** (i.e. the input and output) and the power spectral density of the input, **x** based on the Welch averaged periodogram method:

$$H_{xy}(f) = \frac{S_{xy}(f)}{S_{xx}(f)}$$



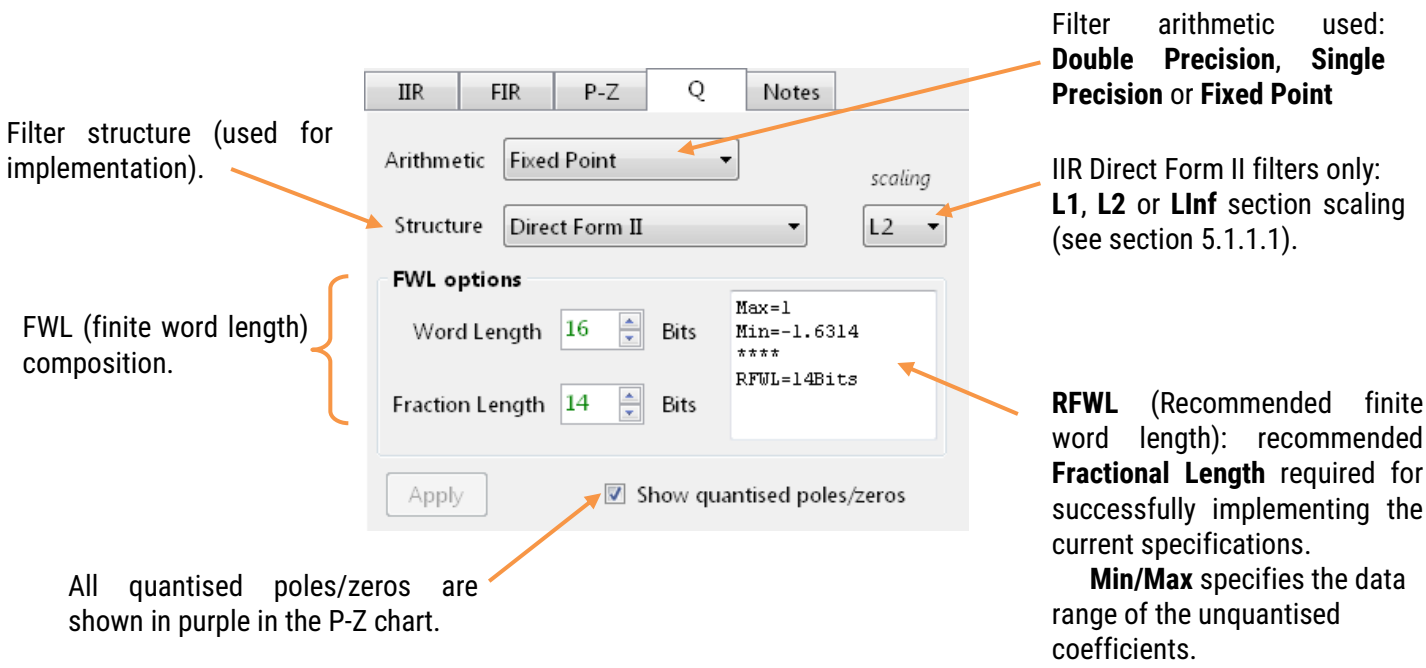
Enabling this functionality automatically sets the frame size to `1500`, enables the white noise signal generator and disables the input spectrum trace (although the data still used in the computation). The averaging is performed over 10 frames using a **Hanning** window (default).

Care should be exercised when interpreting the results, as closely grouped poles/zeros may not appear to match the design specifications. Also, the results should only be interpreted as an estimation based on a window length of `1500` using a **Hanning** Window. Other types of Window functions will give different results, where the following functions are available:

Rectangular	**FlatTop**	**Hamming**
Blackman	**Lanczos**	**Hanning**
Blackman-Harris	**Gaussian**	

5. H1 quantisation options and filter structure

The ASN filter designer provides designers with a rich assortment of quantisation analysis options for H1 filters.

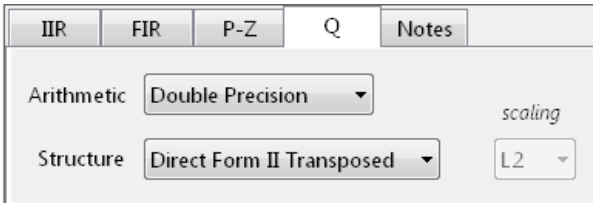


5.1. H1 Filter structures

You may experiment with various H1 filter structures and quickly assess your design's performance with different structures and quantisation settings.

H1 IIR structures:

- Direct Form I
- Direct Form II
- Direct Form II Transposed (default)



H1 FIR structures:

- Direct Form
- Direct Form Transposed (default)

5.1.1. FWL (finite word length)

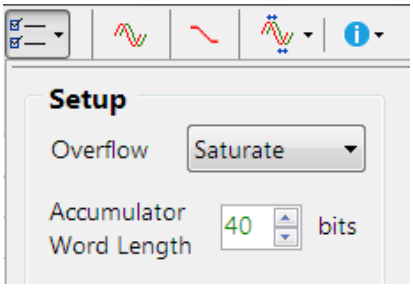
The system word length is split up into its *number of integer bits* and its *number of fractional bits (fractional length)*. Where, the general format is given by:

Q *number of Integer bits.number of fractional bits*

For example, if we assume that all of data values lie within a maximum/minimum range of ±1, we can use **Q0.15** format to represent all of the numbers respectively. Notice that **Q0.15** format is a 16-bit word, comprised of 1 sign bit, with a maximum of $2^{15} - 1 = 32767$ and a minimum of $-2^{15} = -32768$.

Word Lengths of between 8-32bits may be implemented.

Accumulator word length options and overflow rules can be found in the signal analyser setup menu as discussed in section 4.4.



5.1.1.1. Direct Form II scaling

When implementing Direct Form II IIR filters, it is necessary to ensure that the feedback path, *w(n)* will not overflow (see section 2.4.2). The following scaling methods are available:

L1 norm	$G = \sum_{n=0}^{\infty} \lvert w(n) \rvert$	L1 norm assumes that the input is bounded and ensures that regardless of the type of input there will be no overflow. Needless to say, L1 scaling is extreme and should only be used when L2 or L∞ scaling is insufficient.
L2 norm	$G = \left[\sum_{n=0}^{\infty} w^2(n) \right]^{\frac{1}{2}}$	L2 norm places an energy constraint on the input and output transfer function. This scaling scheme offers the smallest scaling factor.
L∞ norm	$G = \max \lvert W(w) \rvert$	L∞ norm ensures that the filter will not overflow when a sine wave is applied.

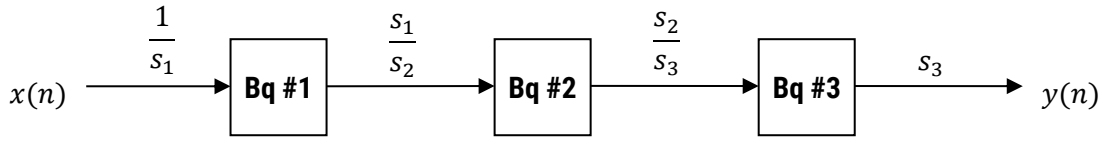
If $|a_k| \le 2$ and $|b_k| \le 2$ then the following difference equations may be used:

$$d(n) = \frac{x(n)}{G} - a_1 w(n-1) - a_2 w(n-2)$$

$$y(n) = G \times \left[b_0 w(n) + b_1 w(n-1) + b_2 w(n-2) \right]$$

5.1.1.1.1. Biquad cascade scaling

A biquad cascade comprised of three biquads is shown below.



The scaling coefficients are given as s_1, s_2 and s_3 respectively. The filter designer tool automatically scales Bq#1's numerator coefficients by $G_1 \times \frac{s_1}{s_2}$, Bq#2's numerator coefficients by $G_2 \times \frac{s_2}{s_3}$ and Bq#3's numerator coefficients by G_3. The input scaling factor, $\frac{1}{s_1}$ and output scaling factor, s_3 are summarised in the filter summary under **Cascade Scaling Factors**. Where, Input actually given as s_1 instead of $\frac{1}{s_1}$ and **Output** is s_3. As a final point, rather than using the exact scaling factors, the values are actually rounded to the power of 2 (i.e. 2,4,8,16 etc.) in order to simply the implementation.

5.1.1.1.2. Example

In order to fully understand the information presented in the ASN filter designer, the following example illustrates the filter coefficients obtained with double precision and with **Q1.14** quantisation.

```
Biquad #1
Gain = 0.022065
B = [ 1.00000000000,  1.42515458311,  1.00000000000]
A = [ 1.00000000000, -1.49439567907,  0.56622636801]

Biquad #2
Gain = 0.059997
B = [ 1.00000000000, -0.21088913424,  1.00000000000]
A = [ 1.00000000000, -1.56831045118,  0.67755833899]

Biquad #3
Gain = 0.122786
B = [ 1.00000000000, -0.77860154757,  1.00000000000]
A = [ 1.00000000000, -1.71966704418,  0.87471907332]
```

double precision

Applying L2 scaling with Q1.14 (note the effects of quantisation on the coefficient values), we obtain ($s_2 = 4$):

```
Cascade Scaling Factors
Input = 8
Output = 8

Biquad #1
Bq = [ 0.04412841797,  0.06286621094,  0.04412841797];  [ 723,  1030,  723]
Aq = [ 1.00000000000, -1.49438476563,  0.56622314453];  [ 16384, -24484,  9277]

Biquad #2
Bq = [ 0.02996826172, -0.00634765625,  0.02996826172];  [ 491, -104,  491]
Aq = [ 1.00000000000, -1.56829833984,  0.67755126953];  [ 16384, -25695,  11101]

Biquad #3
Bq = [ 0.12280273438, -0.09558105469,  0.12280273438];  [ 2012, -1566,  2012]
Aq = [ 1.00000000000, -1.71966552734,  0.87469482422];  [ 16384, -28175,  14331]
```

$$G_1 \times \frac{s_1}{s_2} = 0.022065 \times \frac{8}{4} = 0.0441$$

$$G_2 \times \frac{s_2}{s_3} = 0.05999 \times \frac{4}{8} = 0.0299$$

$$G_3 = 0.1228$$

decimal coefficient value:
ceil $(2^{14} \times 0.02996) = 491$

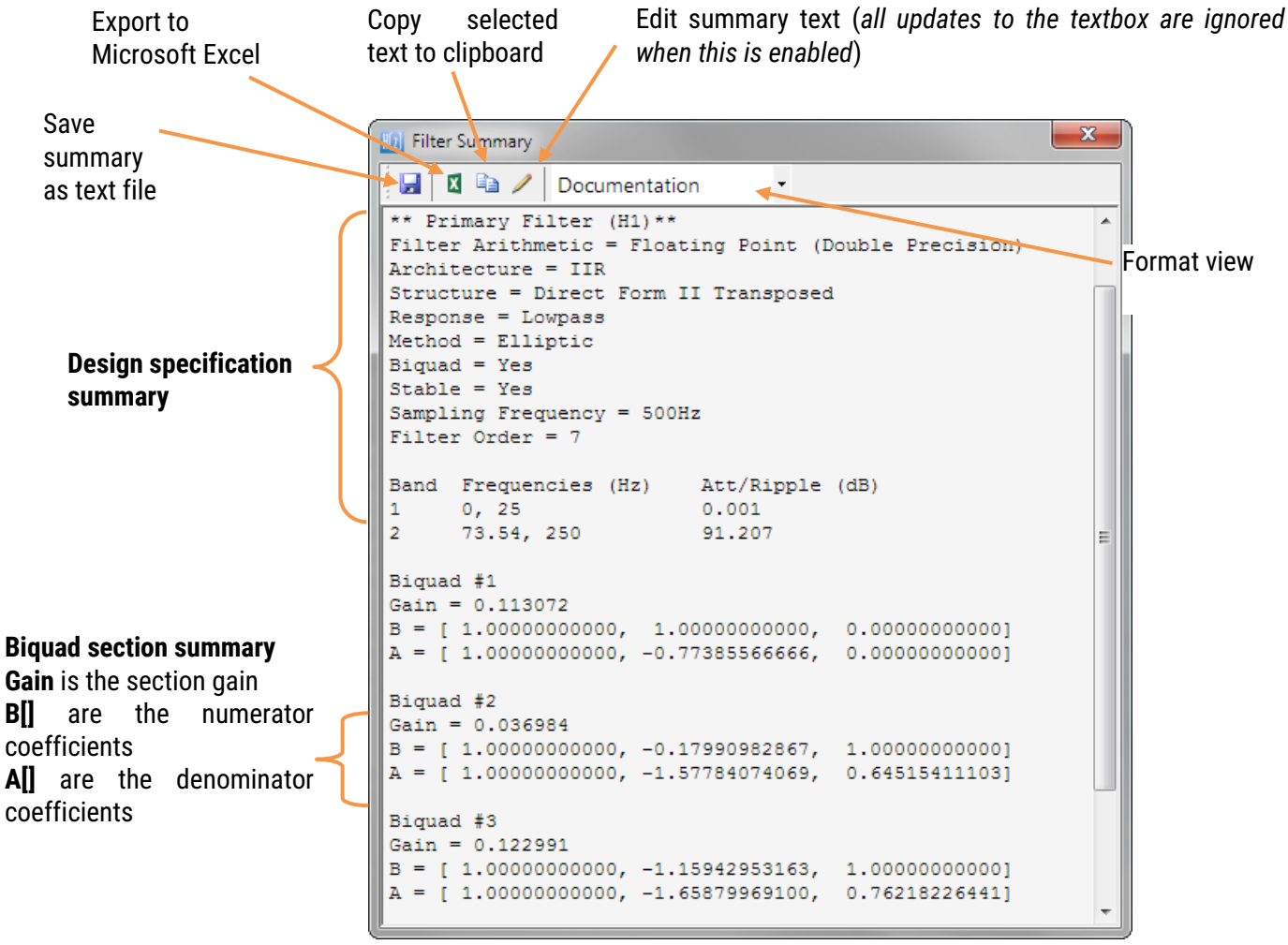
The decimal coefficients may be directly inserted into a fixed-point algorithm for implementation.

5.2. Filter summary

The filter summary presents the designer with a detailed summary of the filter coefficients and technical specifications used for the design. These details may be exported to third party applications, such as Matlab for further analysis. The following export formats are supported:

- Matlab/Octave
- Scilab
- ANSI C

A detailed overview of each framework can be found in section 5.2.4.



If the design is modified via the P-Z editor, the **Response** string changes to **User Defined** and the **Method** string is removed as the design does not adhere to a prototype design method anymore.

In order expedite the design and integration phase with 3rd party design tools, such as Analog Devices' SigmaStudio, registered users may export the filter summary to Microsoft Excel.

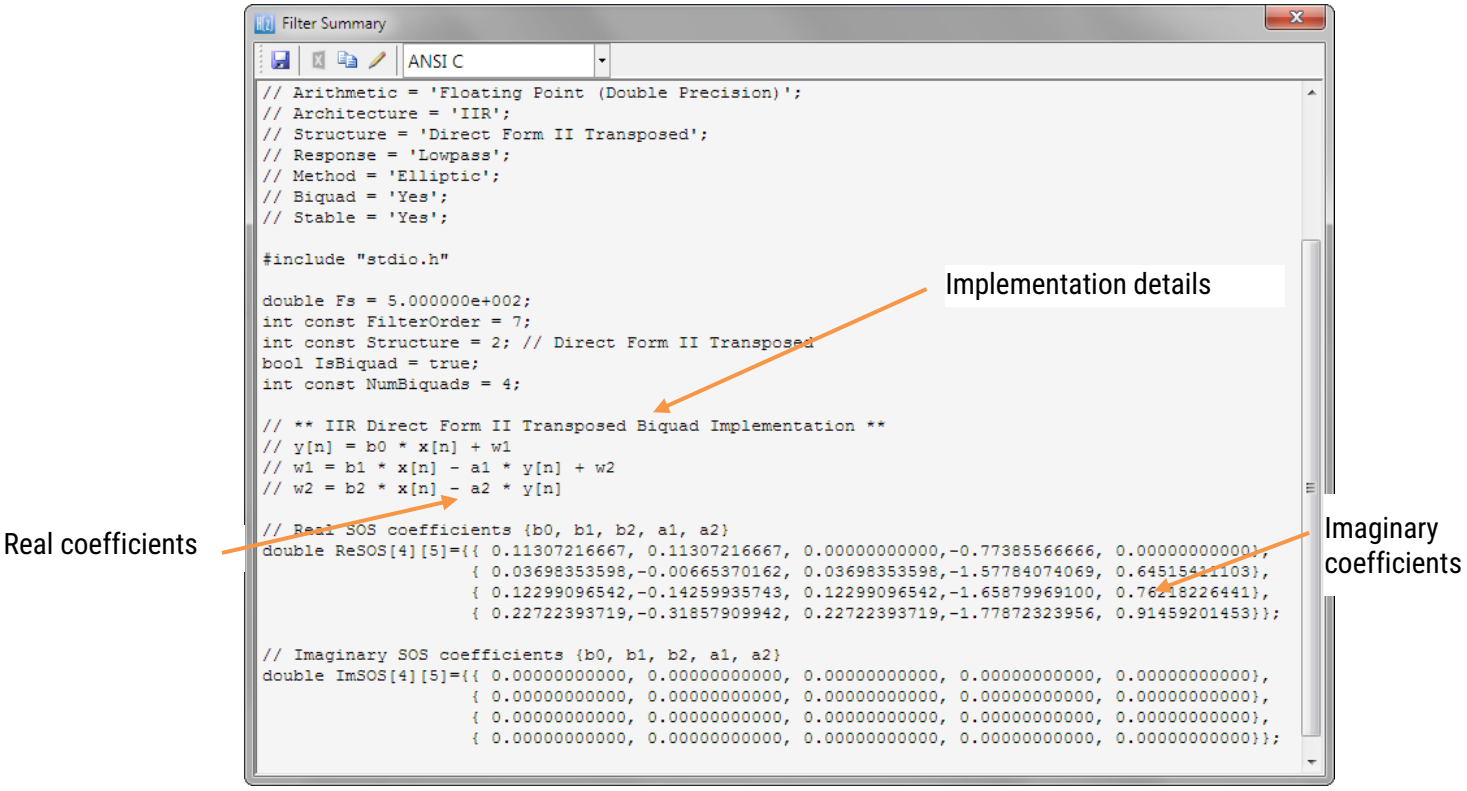
The export feature is only available in *Documentation* mode.

5.2.1. Complex coefficients

The ASN filter designer fully supports the design and analysis of both real and complex coefficient filters. When exporting the coefficients in **ANSI C** format, care should be taken when interpreting the coefficient representation.

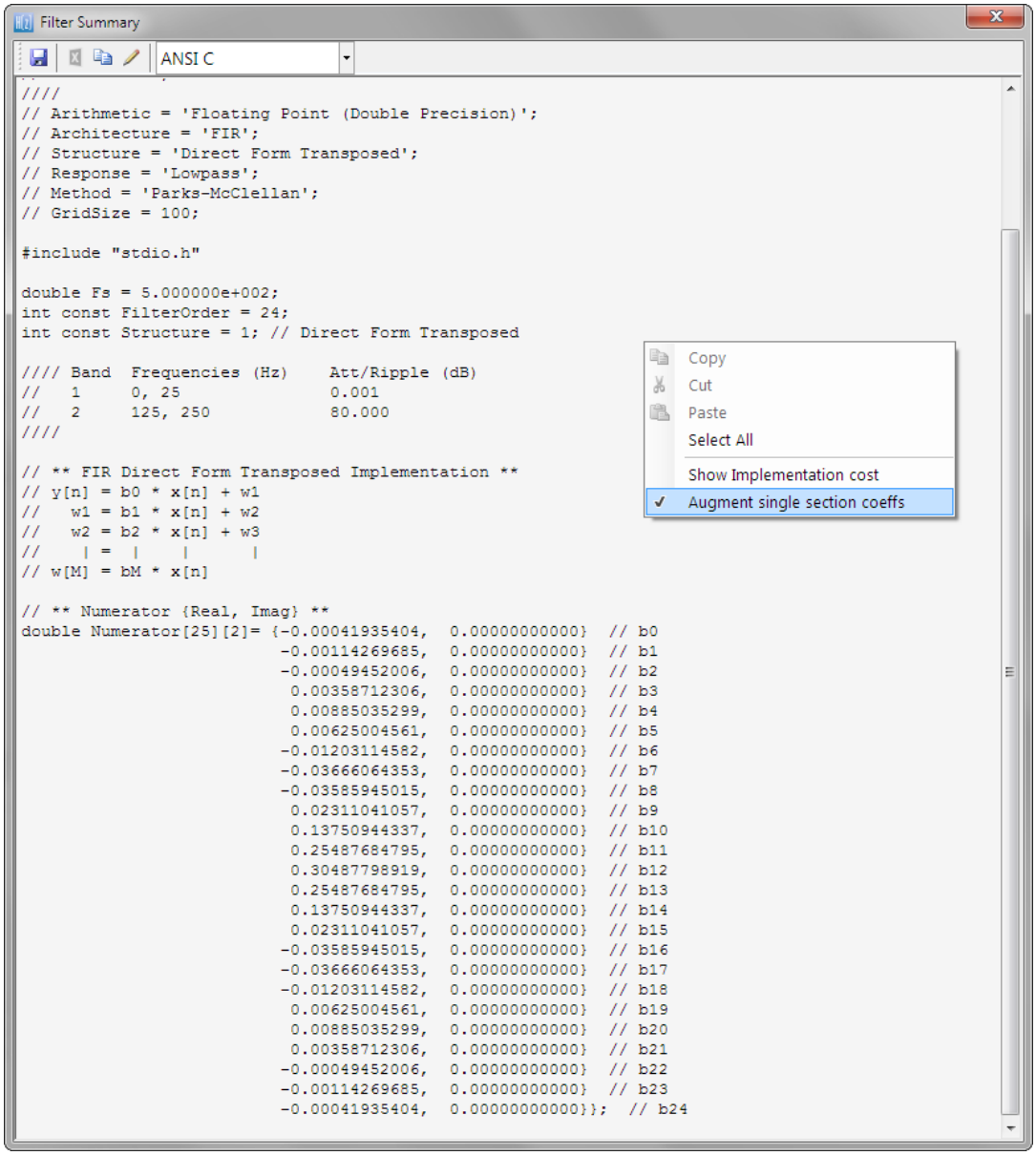
5.2.1.1. IIR biquad

Biquad filter coefficients are summarised as follows, where it can be seen each row of **ReSOS** and **ImSOS** multidimensional array pertains to a single Biquad implementation.



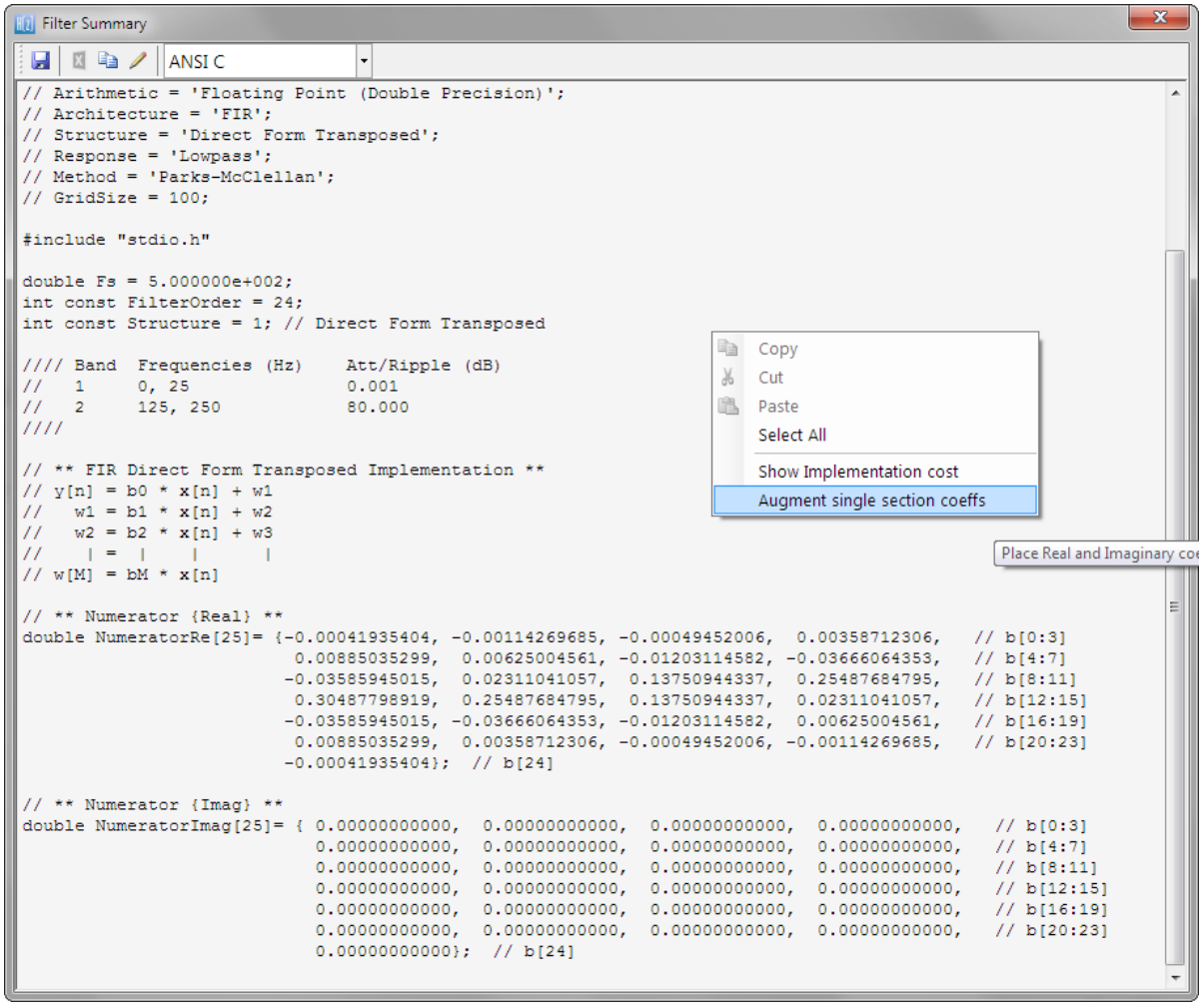
5.2.1.2. Single section IIR and FIR

As per default, single section IIR/FIR real and complex coefficients are bundled into a single multidimensional array, as shown below. This format is compact and is also required for the ANSI C framework (see section 5.2.4).

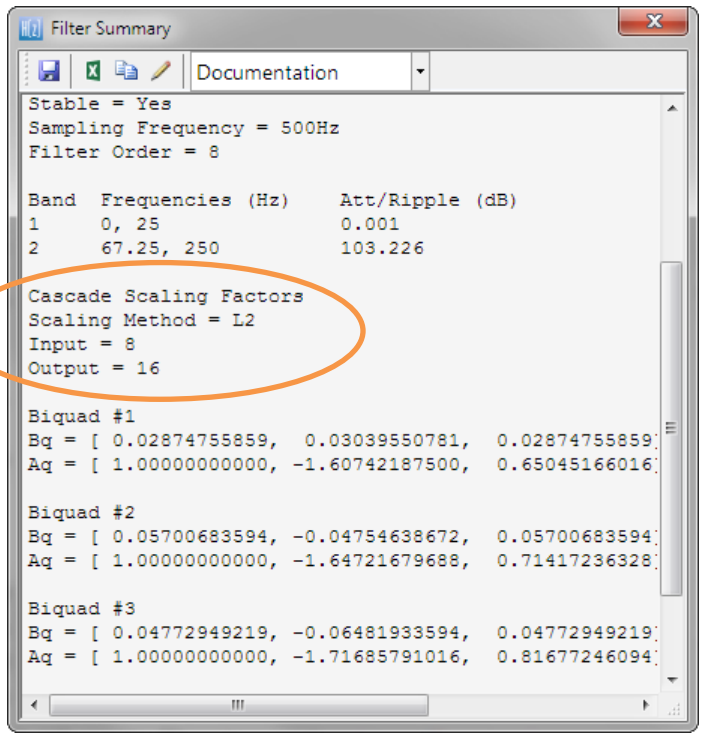



```
////
// Arithmetic = 'Floating Point (Double Precision)';
// Architecture = 'FIR';
// Structure = 'Direct Form Transposed';
// Response = 'Lowpass';
// Method = 'Parks-McClellan';
// GridSize = 100;

#include "stdio.h"

double Fs = 5.000000e+002;
int const FilterOrder = 24;
int const Structure = 1; // Direct Form Transposed

//// Band  Frequencies (Hz)    Att/Ripple (dB)
//   1     0, 25               0.001
//   2     125, 250            80.000
////

// ** FIR Direct Form Transposed Implementation **
// y[n] = b0 * x[n] + w1
//   w1 = b1 * x[n] + w2
//   w2 = b2 * x[n] + w3
//    | =  |    |      |
// w[M] = bM * x[n]

// ** Numerator {Real, Imag} **
double Numerator[25][2]= {-0.00041935404,  0.00000000000}  // b0
                         -0.00114269685,  0.00000000000}  // b1
                         -0.00049452006,  0.00000000000}  // b2
                          0.00358712306,  0.00000000000}  // b3
                          0.00885035299,  0.00000000000}  // b4
                          0.00625004561,  0.00000000000}  // b5
                         -0.01203114582,  0.00000000000}  // b6
                         -0.03666064353,  0.00000000000}  // b7
                         -0.03585945015,  0.00000000000}  // b8
                          0.02311041057,  0.00000000000}  // b9
                          0.13750944337,  0.00000000000}  // b10
                          0.25487684795,  0.00000000000}  // b11
                          0.30487798919,  0.00000000000}  // b12
                          0.25487684795,  0.00000000000}  // b13
                          0.13750944337,  0.00000000000}  // b14
                          0.02311041057,  0.00000000000}  // b15
                         -0.03585945015,  0.00000000000}  // b16
                         -0.03666064353,  0.00000000000}  // b17
                         -0.01203114582,  0.00000000000}  // b18
                          0.00625004561,  0.00000000000}  // b19
                          0.00885035299,  0.00000000000}  // b20
                          0.00358712306,  0.00000000000}  // b21
                         -0.00049452006,  0.00000000000}  // b22
                         -0.00114269685,  0.00000000000}  // b23
                         -0.00041935404,  0.00000000000}};  // b24
```

Checking the **Augment single section coeffs** option in the right mouse button menu enables designers to set the coefficients format to the compact multidimensional array format. However, if designing real filters (i.e. complex values are not required), you may separate the coefficients into separate arrays by unchecking the **Augment single section coeffs** option.



5.2.2. Fixed point

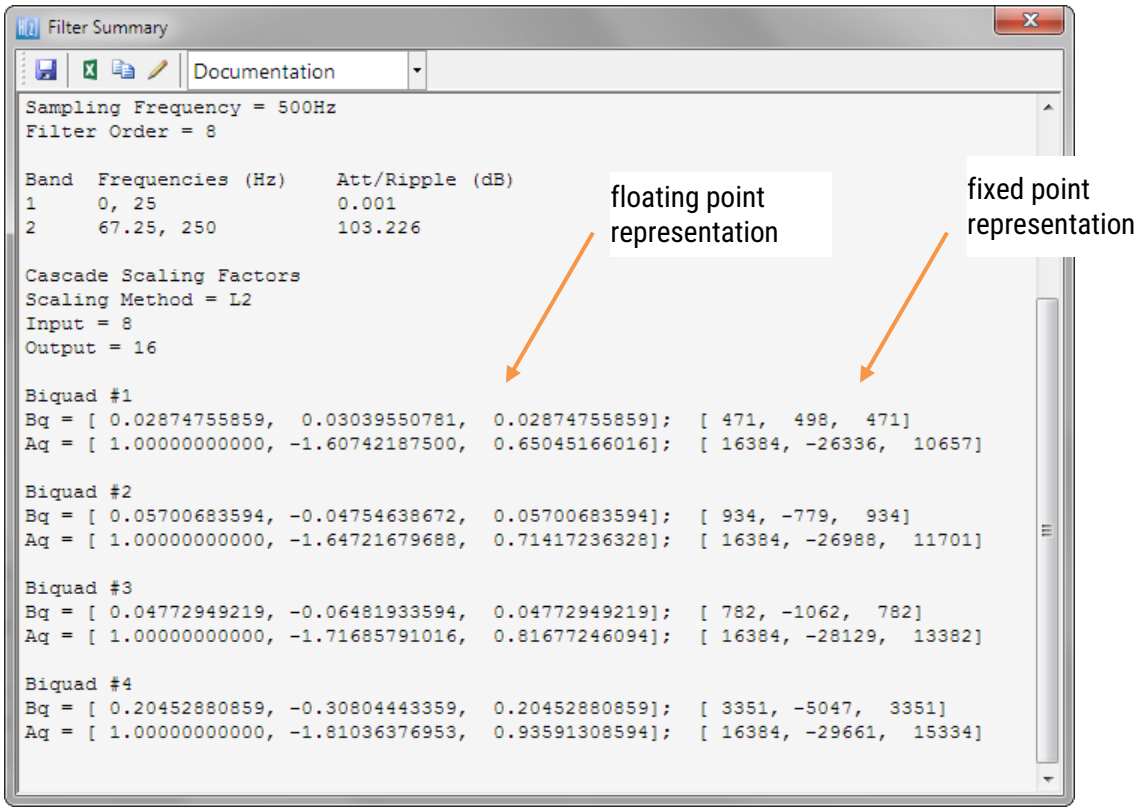


Rather than overwhelming the designer with information, the filter summary strives to present the designer with a concise fixed point summary.

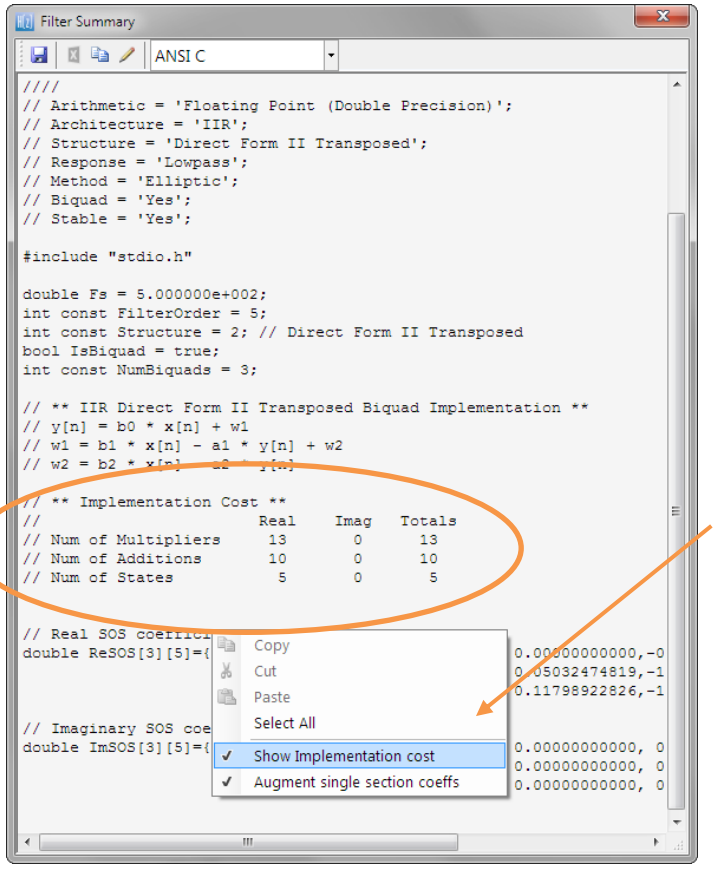
When implementing IIR filters using the direct form II architecture, scaling is automatically included into the section number coefficients. The only two pieces of scaling information needed for implementation are the **Input** and **Output Cascade Scaling Factors.** Where, these factors are needed to scale the input and output results respectively.

Input and **Output Cascade Scaling Factors** are discussed depth in section 5.1.1.1.1.

The quantised coefficient values are presented in two formats: fixed point and floating point, as shown below.



5.2.3. Implementation cost summary



ANSI C only

In order to assist with implementation in embedded devices, such as an FPGA or a DSP, an implementation cost is available. This summary gives the designer a quick overview of the number of summers, multiplications and state variables needed for implementing the designed filter.

You may show or hide the implementation cost summary by setting the option in the context menu (right mouse button).

5.2.4. Support for 3rd party software development frameworks

Comprehensive royalty free software development frameworks for Matlab, Scilab and ANSI C (including examples) can be found in the tool's installation directory. The ANSI C software development framework is especially useful for fixed point implementation, whereby the tool produces a C header file for direct use in your application.

Due to the flexibility of the data formats, developers may easily edit the files for use with other third party development applications, such as LabView, Excel etc.

Directory name	Description
\Matlab	Matlab filter framework. The following three files constitute the framework: ASNFDMatlabFilterData.m ASNFDMatlabImport.m ASNFDFilter.m **Demo files** RMSmeterDemo.m — An RMS amplitude meter demo based on the Hilbert transform. EMGDataDemo.m — A three filter EMG biomedical example. ASNFDMatlabDemo.m — A general purpose demo, to be used as an example for your applications.
\Scilab	Scilab filter framework. The following three files constitute the framework: ASNFDScilabFilterData.sce ASNFDScilabImport.sce ASNFDFilter.sce **Demo files** RMSmeterDemo.sce — An RMS amplitude meter demo based on the Hilbert transform. EMGDataDemo.sce — A three filter EMG biomedical example. ASNFDScilabDemo.sce — A general purpose demo, to be used as an example for your applications.
\ANSIC\IIR	ANSI C IIR biquad filter framework. The following two files constitute the framework: ANSIC_IIR_DFF.h ANSIC_IIR_DFF.cpp FilterSummary.h — An example IIR exported from the ASNFD.
\ANSIC\FIR	ANSI C FIR filter framework. The following two files constitute the framework: ANSIC_FIR_DFF.h ANSIC_FIR_DFF.cpp FilterSummary.h — An example FIR exported from the ASNFD.

6. Other options: Project files and Design notes

The ASN filter designer allows licensed users to save any project design notes as part of the project file, providing a powerful documentation solution, suitable for peer review and/or project handover. The tool has been designed in order to allow non-licensed (demo) users to view the design notes and filter frequency response of a design created with either the educational or professional version, which is ideal for students and demonstrating your design to clients.

6.1. Opening project files

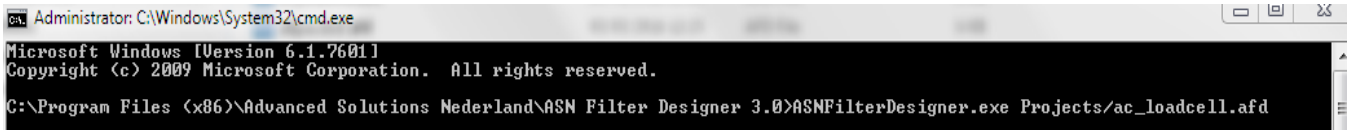
All versions of the software allow you to load a project file. For the demo version, this has the added advantage of allowing you to load a project file created in the either the professional or educational version for evaluation purposes.

Although the filter's frequency response may always be analysed for all versions, the signal generator (see section 4) places a limit on the maximum licensed filter order. This usually impacts users of the demo version, as a project file created in either the educational or professional version may be loaded, but it may not always be possible to use the signal generator.

6.1.1. Opening project files from Window's command line prompt interface

Project files (`*.afd`) may be loaded into the ASN filter designer via Window's command line prompt interface using the following syntax:

```
ASNFilterDesigner.exe  <filename>
```



6.1.2. Drag and drop

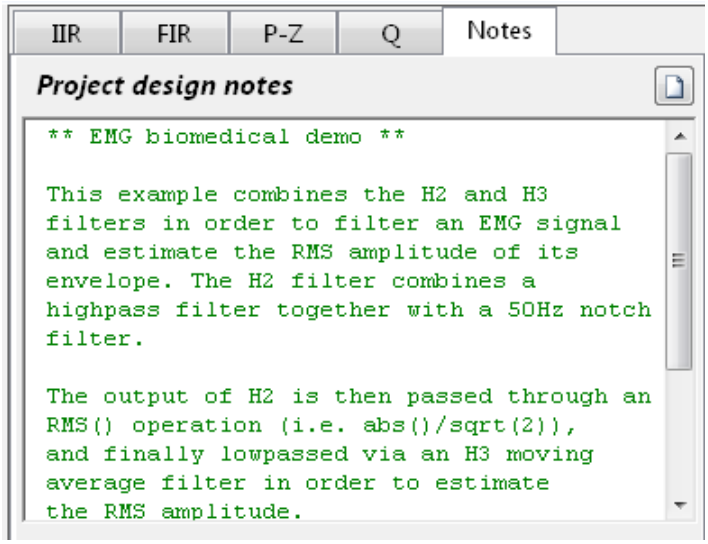
As with many Windows based applications, dragging and dropping an `afd` project file onto the main chart will automatically open it.

6.2. Saving project files

Only licensed (i.e. educational and professional) users of the tool may create project files. Clicking on the arrow will allow you to **Save As**

6.3. Project design notes

Any project design notes may be entered in the **Project design notes** textbox. Formatted text, such bold and italics is currently not supported, but hyperlinks are supported and are a useful means of helping document and maintain a design.



The project design notes are automatically saved when saving a project file (licensed users only), and automatically updated when a project file is loaded.

Part II

Bespoke filter design and the H2 Filter

7. Introduction

The H1 (primary) filters considered in Part I are all designed via standard prototype methods, such as Butterworth, Chebyshev for IIR filters and Parks-McClellan for FIR filters. Although these design methods are adequate for many applications, they are limited in their flexibility. As an example, consider the transfer function of an IIR notch filter:

$$H(z) = \frac{1 - 2\cos w_c z^{-1} + z^{-2}}{1 - 2r\cos w_c z^{-1} + r^2 z^{-2}}$$

where, $w_c = \frac{2\pi f_c}{fs}$ controls the centre frequency, f_c of the notch, and r controls the bandwidth of the notch. Clearly this cannot be implemented with a standard IIR prototype.

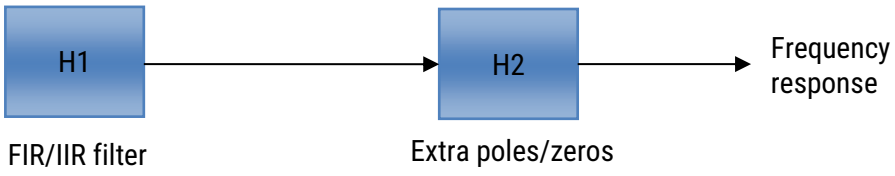
The ASN Filter designer offers designers two powerful methods for designing bespoke (specialised) filters:

- **P-Z editor**: A fully interactive pole-zero editor allowing designers to zoom, pan and graphically fine-tune designs to their exact requirements. The corresponding frequency response is updated in real-time allowing for instant evaluation of the new pole-zero positions. The zooming and panning feature is also available in the pole-zero chart, allowing designers to easily fine tune the pole-zero positions with the mouse and see the effects in real-time on the frequency response chart.

- **ASN Filter Script**: A scripting language supporting over 40 scientific commands, provides designers with a familiar and powerful programming language, while at the same time allowing them to implement complex symbolic mathematical expressions for their filtering applications. The scripting language interface offers the unique and powerful ability to modify parameters on the fly with the so called interface variables, allowing for real-time updates of the resulting frequency response. This has the advantage of allowing the designer to see how the coefficients of the symbolic transfer function expression affect the frequency response and the filter's time domain dynamic performance.

7.1. The primary (H1) and secondary (H2) filter

ASN filter designer allows designers to add extra poles and zeros to any IIR or FIR filter via the P-Z editor. In order to facilitate this, the main FIR/IIR filter is assigned to the *primary filter, H1* and any extra pole/zeros are added to a *secondary filter block, H2*. The H2 filter block implements the filter as a **Direct Form II Transposed** single section IIR or a **Transposed Direct Form** FIR (if no poles are present). Notice that this degree of flexibility has the advantage of assigning poles to an FIR primary filter.



It should be noted that a direct form (single section) implementation may become problematic (due to numerical stability issues) for higher filter orders, especially when poles are near to the unit circle.

7.2. Design methods and customisation

All H1 (primary) filters are designed via standard prototype methods, such as Butterworth, Chebyshev for IIR filters and Parks-McClellan for FIR filters. The pole-zero positions of these 'standard filters' may be modified by the user via the P-Z editor in order to customise the design.

As discussed in section 8.2.2, the P-Z editor allows designers to combine and H1 and H2 filters in order to accommodate more advanced design requirements. This is especially useful for adding extra characteristics to a standard filter with minimal effort, for example, adding an extra null (zero) to a lowpass filter. H2 filters may be specified by the P-Z editor (i.e. manually adding poles and zeros one-by-one to the z-plane), or by the symbolic math script language (see section 8.2.3). Where, the latter allows designers to specify and experiment with the H2 transfer function symbolically.

7.3. Errors in high order polynomials

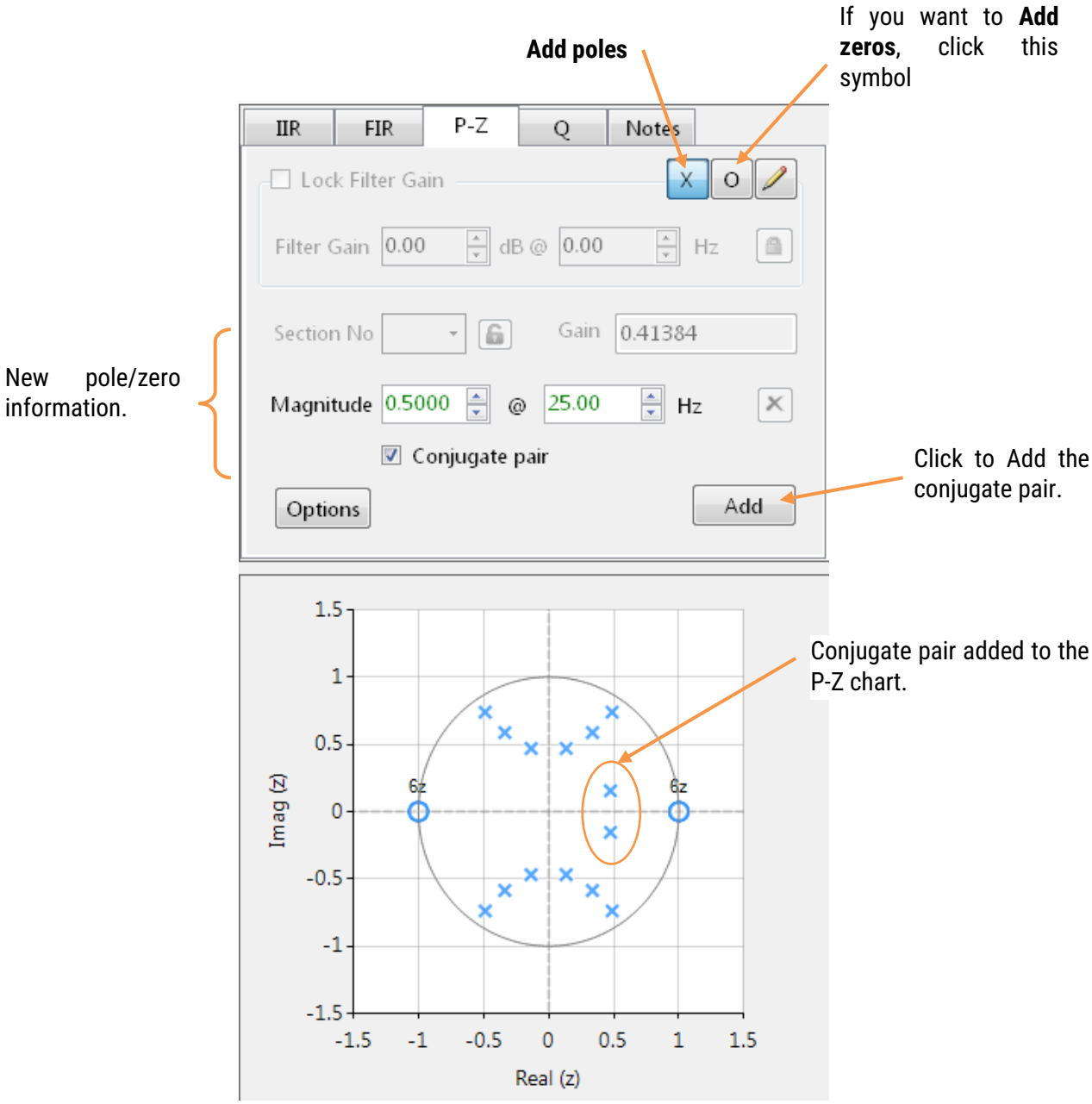
The tool will for FIR filters and the filter script use the given **Num** and **Den** polynomials for computation. However, if these positions are modified via the P-Z editor, they will be handled via the roots-to-poly algorithm which will attempt to reconstruct the polynomial from the presented roots using double precision arithmetic. For lower orders this will generally result in an almost identical polynomial, but as a consequence of the errors inherent to the root finding algorithm, higher order polynomials (> 60 or so) may significantly deviate from the ideal result.

8. P-Z editor

The P-Z editor was introduced in section 3 for editing the properties of an H1 filter. However, the editor also allows designers to add poles and zeros to a design which are implemented in the H2 filter.

Example

The following example illustrates the ease at which a conjugate pole pair can be added to the H2 filter:



Use **Edit mode** to edit the properties of the new conjugate pole pair.

8.1. Section number and section lock

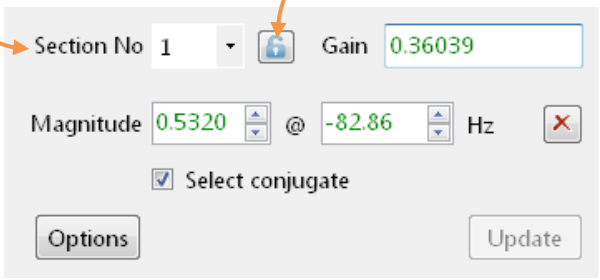
Section number

This allows you to highlight the pole and zeros of a specific section in the H1 (the primary filter) or the pole-zeros of the H2 (secondary) filter:

- For FIR filters or single section IIR filters: the section number will always be equal to **1**.
- For biquad IIR filters: this will be a list of all the biquad sections in the filter cascade.
- The secondary filter (H2): is represented as **H2**.

Section lock

Clicking on the section lock, allows you to focus on a specific section by highlighting all of the poles-zeros of the selected section number and minimising the rest.

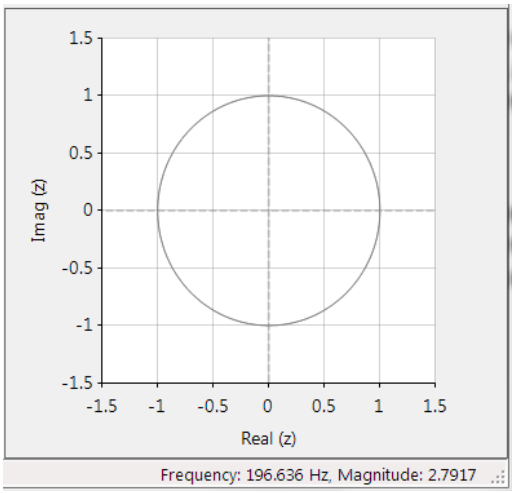


8.2. Options

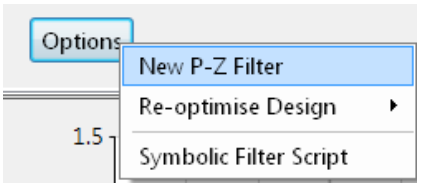
The options menu extends on the functionality of a simple pole-zero editor by allowing designers to design and experiment with any combination of poles and/or zeros of their choice. There are three options as discussed below:

8.2.1. New P-Z filter

The new pole-zero filter option deletes all poles and zeros from the design, and in essence is a blank sheet.



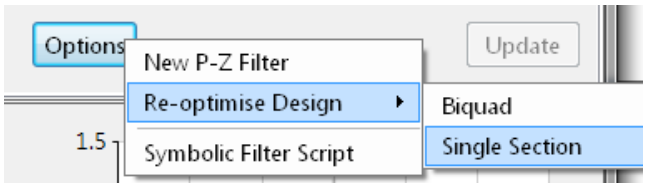
You may add pole and zeros to the design as required via the **Add pole** and **Add zero** options, as discussed at the beginning of this section. Where, all new poles and zeros are added to the H2 filter only.



The New P-Z filter option is especially useful for classroom examples, whereby the effects of moving a single pole or zero around the z-plane and the resulting frequency response can be seen in real-time. In essence, students can graphically see the effects of a pole/zero on the overall frequency response.

8.2.2. Re-optimise design

The re-optimise design option allows for the analysis and re-optimisation of all H1 and H2 poles and zeros into an H1 filter. This is especially useful for IIR biquads, as any extra poles/zeros that may have been added to H2 will be analysed and allocated to the most suitable biquad. Notice that the sub-menu allows you to choose between a single section or biquad.



In the event that you have added poles to an FIR filter, this option allows you to convert your FIR/IIR design into an H1 IIR filter.

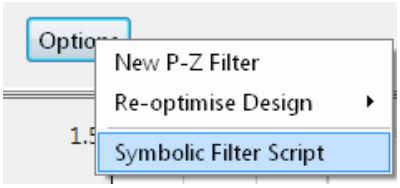
The IIR biquad section optimisation algorithm groups conjugate pole pairs with their closest conjugate zero pairs - where the conjugate pair that is closest to the unit circle is placed at the end of the filter cascade. The optimisation method also attempts to group any non-conjugate poles/zeros to any remaining conjugate pole/zero pairs.

Care should be taken when optimising a design with the symbolic filter script (see section 8.2.3), as the optimisation algorithm decomposes H2's **Num** and **Den** polynomials into their poles and zeros and then combines them with H1's poles and zeros. This optimisation may lead to a slightly different frequency response than the original design for high order polynomials due to errors in the root finding algorithm and different pole-zero pairing combinations.

It is advised in the case of high order polynomials to first fine tune the pole-zero positions in the P-Z editor before applying re-optimisation.

8.2.3. ASN Filter Script

The third and final option provides designers with a powerful symbolic math scripting language.



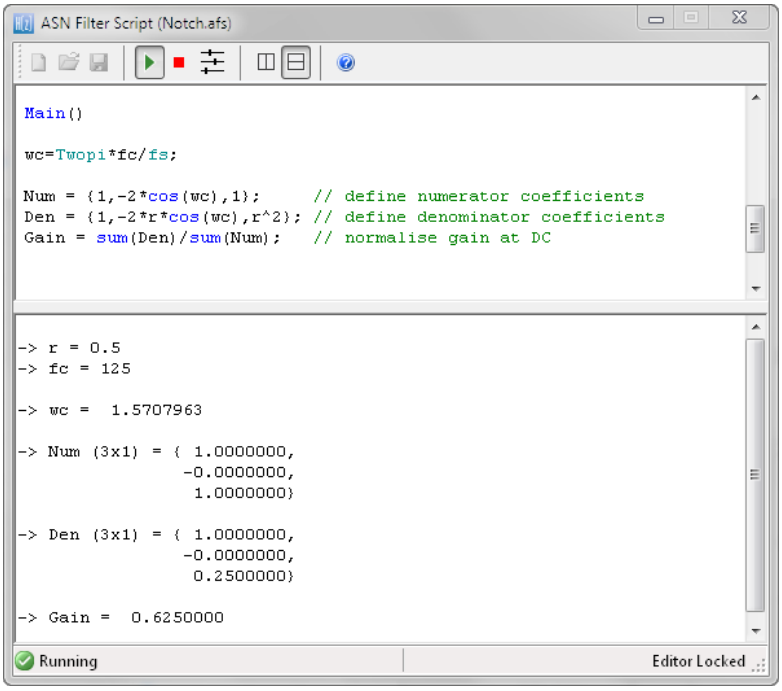
The scripting language supports over 40 scientific commands and allows you to implement complex symbolic mathematical expressions for your filtering applications. The scripting language offers the unique and powerful ability to modify parameters on the fly with the so called interface variables, allowing for real-time updates of the resulting frequency response.

The H2 filter implements the **Num** and **Den** polynomials as defined in the filter script, rather than a re-construction of the roots presented in the P-Z chart. This is particularly useful for high order FIR filters, as no errors are introduced from the root finding algorithm.

However, in the event that any modifications made to the pole-zero positions via the P-Z editor, the tool will automatically re-construct H1 and H2's polynomials by calling the roots-to-poly function.

Example

Revisiting the transfer function presented at the beginning of Part II, we see that almost any symbolic mathematical transfer function can be easily implemented in the Filter script language, as shown below.



$$H(z) = \frac{1 - 2\cos w_c z^{-1} + z^{-2}}{1 - 2r\cos w_c z^{-1} + r^2 z^{-2}}$$

where, $w_c = \frac{2\pi f_c}{fs}$ controls the centre frequency, f_c of the notch, and r controls the bandwidth of the notch.

Setting $r = 0.5$ and $fc = 125$

Please see document ASN15-DOC002 for a complete explanation of the scripting language interface, including detailed practical examples.

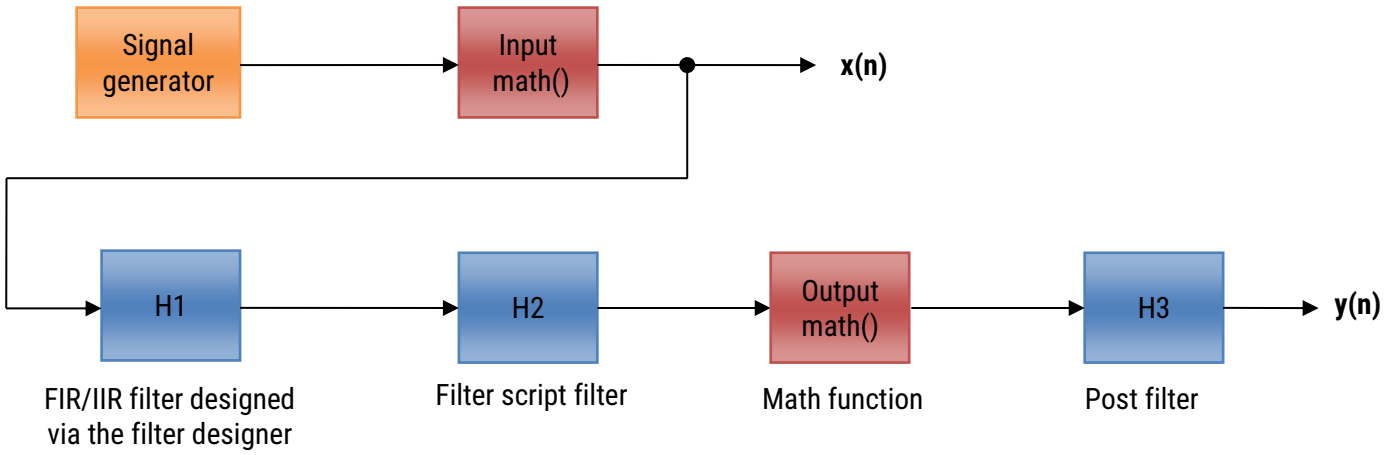
9. H1, H2, H3 filters and the signal analyser

The signal analyser GUI is comprised of a time/frequency domain analyser and a signal generator. The GUI allows designers to explore the time and frequency characteristics of both H1 and H2 type filters for various types of quantisation and inputs. However, when evaluating a fixed point design, only H1 filters may be implemented.

A third filter H3 is also available for post filtering operations, as discussed below in section 9.1.

9.1. H3 post filtering

The signal analyser implements an extra post filter, H3. Unlike the H1 and H2 filters, the H3 filter is *always lowpass*[1] and is preceded by an optional mathematical function operation (i.e. **Abs, Angle, Ln, RMS, Sqr or Sqrt**). The complete filtering chain is shown below together with the signal generator and the input/output math function blocks.

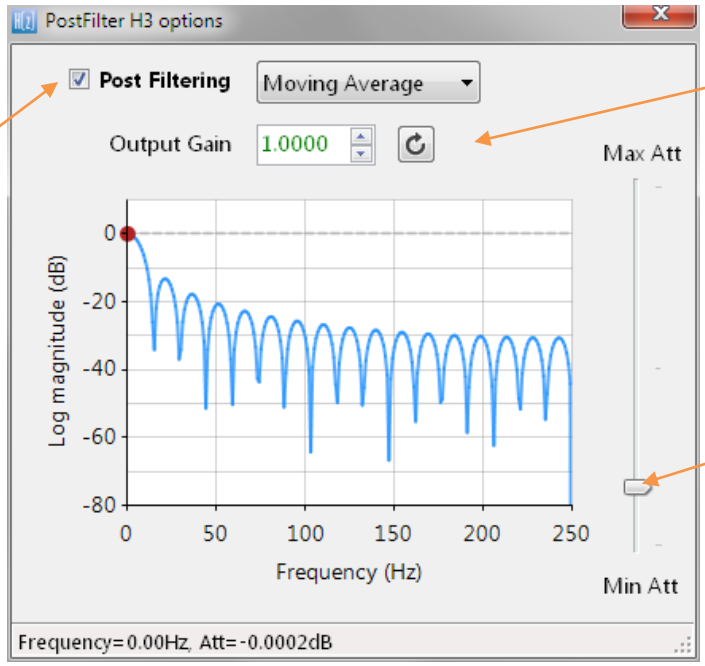


9.1.1. Types of H3 filters

The following three filters are supported:

Type	Transfer function	Gain at DC	Order
IIR	$H_3(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 + 2\alpha z^{-1} + \alpha^2 z^{-2}}$	$\frac{1 + 2\alpha + \alpha^2}{4}$	2
FIR	$H_3(z) = 1 + z^{-1} + z^{-2} \ldots + z^{-M}$	$\frac{1}{(M+1)}$	1-200
Median	*data window*	-	3-195

[1] The median filter is actually a non-linear noise reduction technique.



Enable/disable post filtering.

The post filter is disabled by default.

The output gain (range: `0.1-10`) proceeds the H3 filter, and is useful for fine tuning the output amplitude.

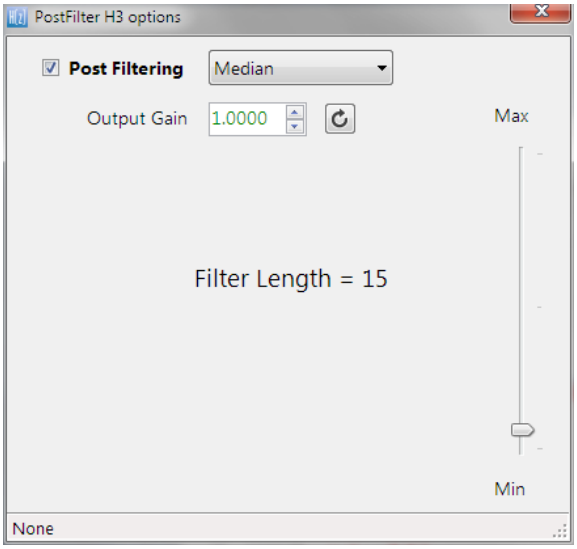
Adjust the slider to change alpha (IIR), or the filter length (FIR/Median)

NB. Unlike the main chart, no zooming functionality is provided and panning is currently limited to the Y-Axis.

As the H3 filter does not form part of the main filter designer, it is *always implemented* using a Direct Form II Transposed (IIR) or Direct Form Transposed (FIR) structure with double precision arithmetic.

9.1.1.1. The median filter

Median filters are a class of non-linear noise reduction filters, which are very good at removing spikes and retaining the sharp edges of signals. The Median filter implemented with the tool does not use a filter structure as such, and just simply computes the median of the data within its buffer (window) using double precision arithmetic.

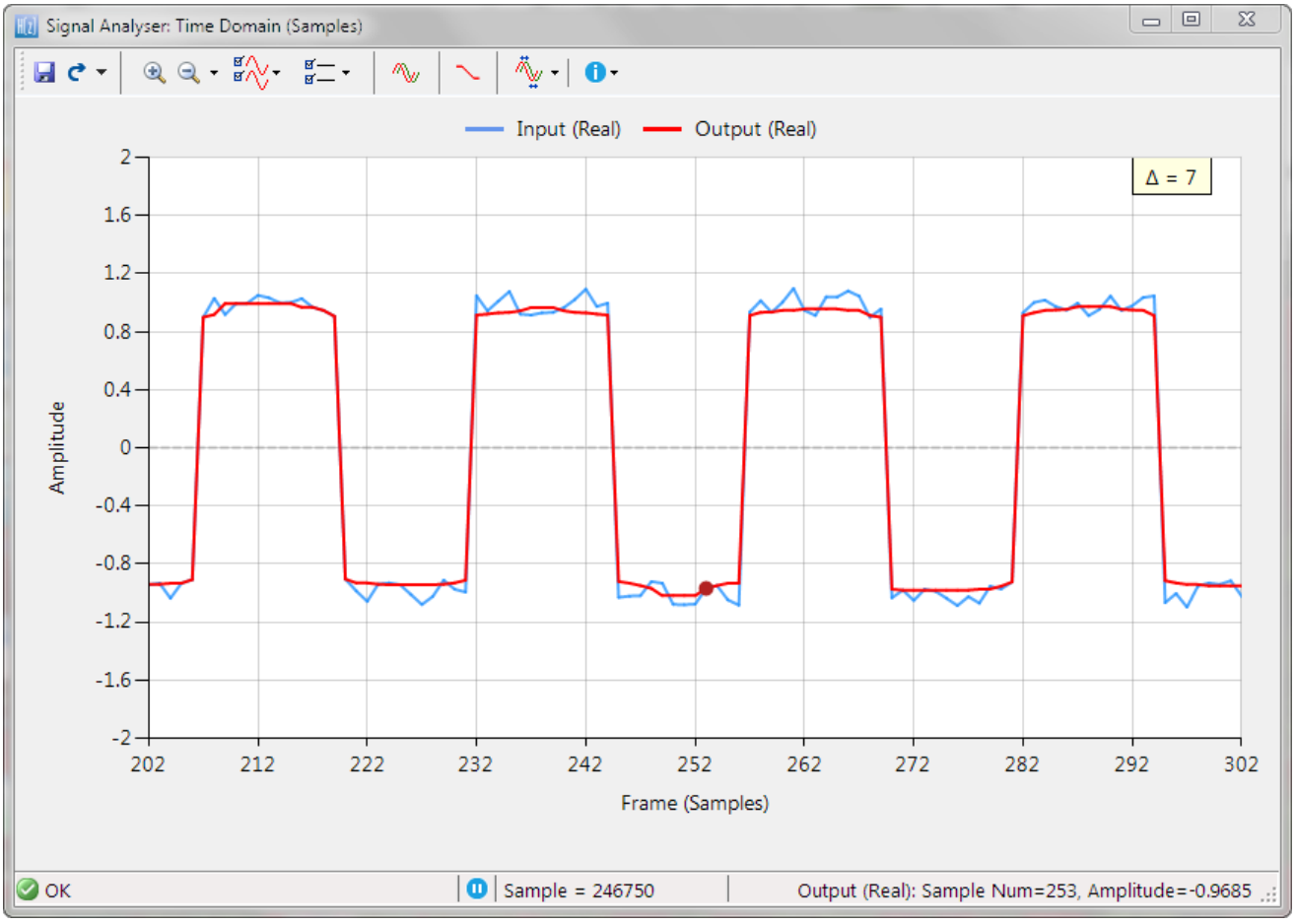


You may use the delay waveform shifting function (as discussed in section 4.3.1.1) by setting the delay to `(Filter Length - 1)/2` samples. Depending on the filter length, this allows designers to align the input and filtered output data streams respectively.

If you just want to implement a median filter, delete the H1 and H2 filters by using **New P-Z filter**.

Time aligned median filtering Example

The following example demonstrates filtering perform with a median filter of **Length** `15` on a 20Hz sinewave sampled at 500Hz. The **Input Delay** has been set to `7`, which aligns the datasets respectively.



Document Revision Status

Rev.	Description	Date
1	Document released.	26/06/2015
2	Updated section 1.1.	09/07/2015
3	Updated sections 5 and 4.5 for new functionality.	15/09/2015
4	Updated document for version 3.0	06/01/2016
5	Added Direct Form II scaling and general improvements.	25/01/2016
6	Updated document for version 3.0.2.	22/02/2016
7	Updated document layout and text for version 3.1.0	26/05/2016
8	Document reviewed and updated.	08/06/2016
9	Document updated for version 3.1.2.	16/11/2016
10	Document updated for version 3.1.4.	07/03/2017