Designing and implementing biquad IIR filters with the ASN Filter Designer: a tutorial review

ADVANCED SOLUTIONS NEDERLAND

Application note (ASN-AN021) February, 2016 (Rev 3)

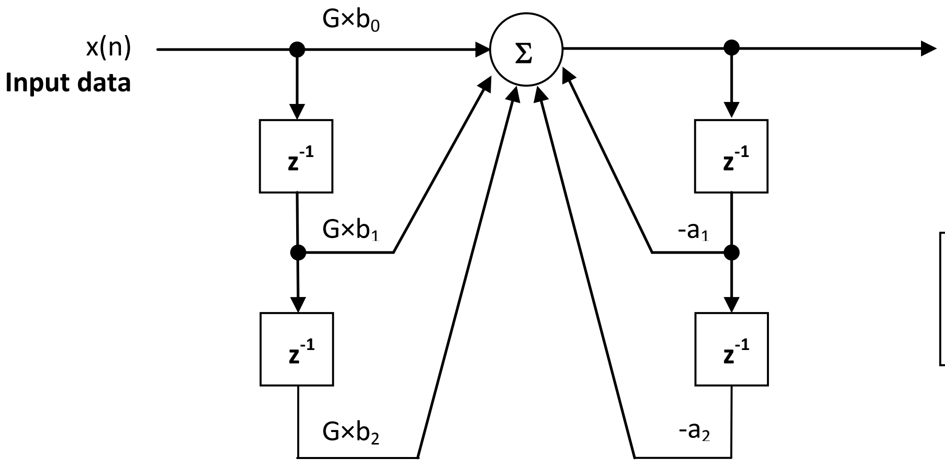
SYNOPSIS

Infinite impulse response (IIR) filters are useful for a variety of signal processing applications, including noise removal and speech coding. Although several practical implementations for the IIR exist, the Direct form II Transposed structure offers the best numerical accuracy for floating point implementation. However, when considering fixed point implementation on a micro-controller or DSP, the Direct Form I structure is considered to be the best choice by virtue of its large accumulator that accommodates any intermediate overflows.

This application note specially addresses IIR biquad filter design and implementation with the ASN filter designer for both floating point and fixed point applications. The discussion is also supported by Matlab scripts allowing the reader to experiment with the concepts presented herein.

INTRODUCTION

The IIR filter implementation discussed herein is said to be 'biquad', since it has two poles and two zeros as illustrated below in Figure 1. The biquad implementation is particularly useful for fixed point implementations, as the effects of quantization and numerical stability are minimised. 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 quantised poles are always inside the unit circle[1].



$$\frac{Y(z)}{X(z)} = G \cdot \frac{\left(b_0 + b_1 z^{-1} + b_2 z^{-2}\right)}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Figure 1 - Direct Form I (biquad) IIR filter realization.

[1] The level of this application note assumes that the reader has a firm grasp of digital filtering and z-transform theory.

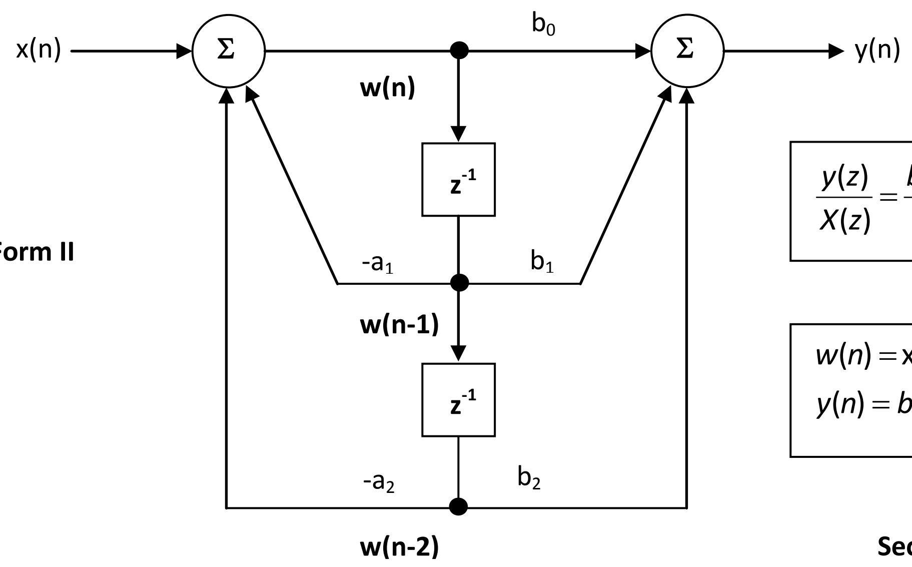
Analysing Figure 1, it can be seen that the biquad structure is comprised of two feedback paths (scaled by a_1 and a_2), two feed forward paths (scaled by b_1 and b_2) and a section gain, G. Thus, the filtering operation of Figure 1 can be summarized by the following simple recursive equation:

$$y(n) = G(b_0x(n) + b_1x(n-1) + b_2x(n-2)) - a_1y(n-1) - a_2y(n-2)$$

Analyzing the equation, notice that the biquad implementation only requires four additions (requiring only one accumulator) and five multiplications, which can be easily accommodated on any modern micro-controller/DSP. The section gain, G may also be pre-multiplied with the forward path coefficients before implementation.

1.1. Floating point implementation

When implementing a filter in floating point (i.e. using double or single arithmetic) Direct Form II structures are considered to be a better choice than the Direct form I structure. The `Direct Form II Transposed` structure is considered the most numerically accurate for floating point implementation, as the undesirable effects of numerical swamping are minimised as seen by analysing the difference equations.

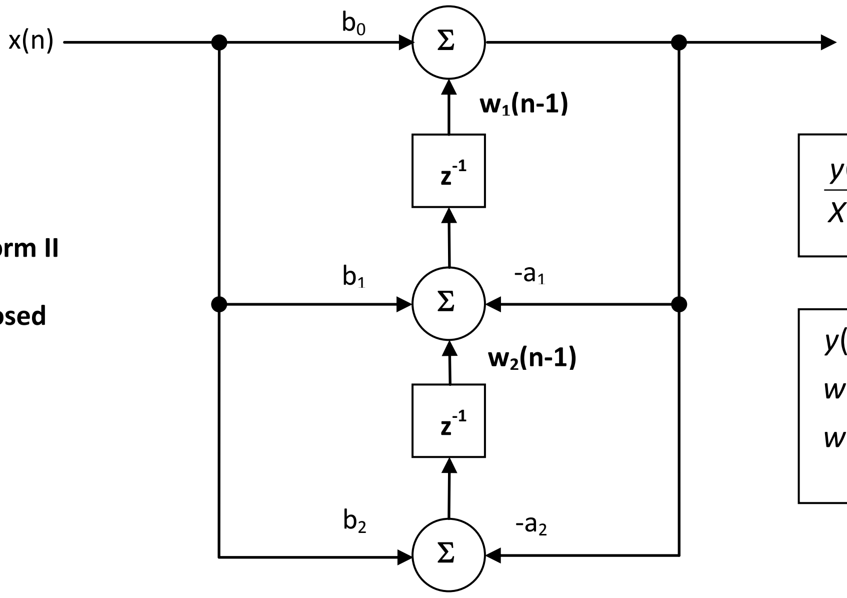


Direct Form II

$$\frac{y(z)}{X(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}$$

$$w(n) = x(n) - a_1w(n-1) - a_2w(n-2)$$
$$y(n) = b_0w(n) + b_1w(n-1) + b_2w(n-2)$$

Section gain, G pre-multiplied with the forward path coefficients.



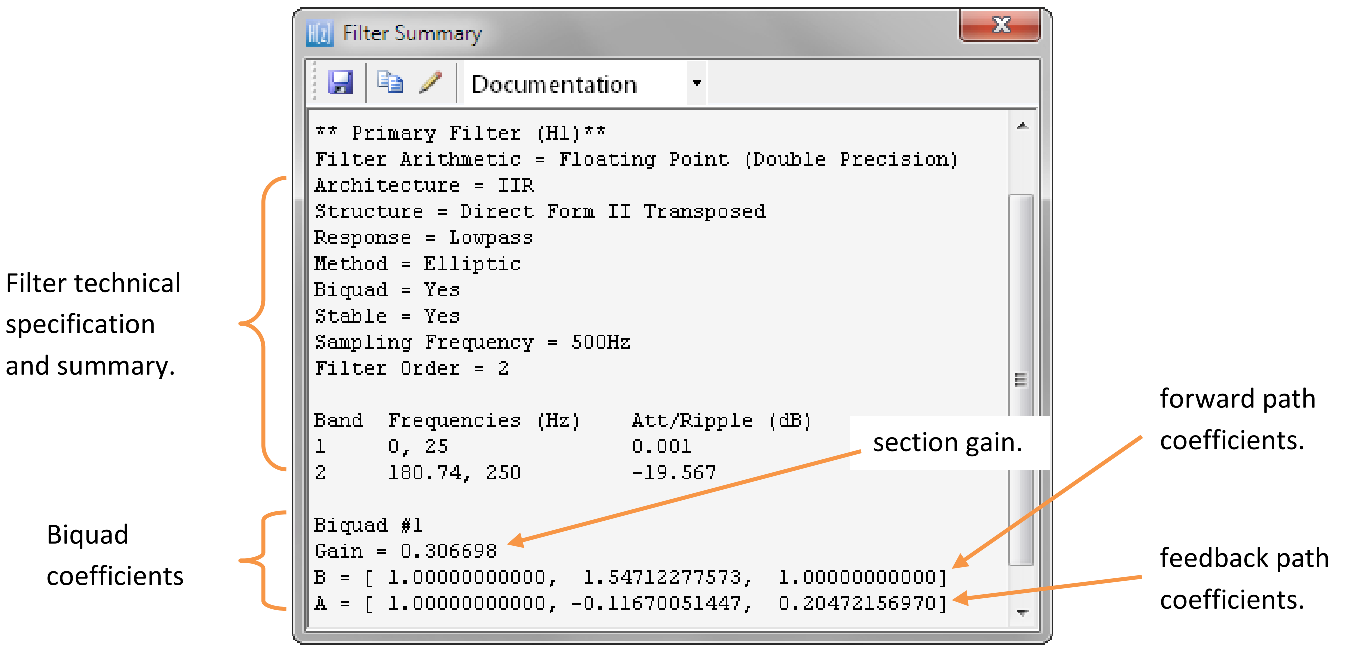
Direct Form II

Transposed

$$\frac{y(z)}{X(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}$$

$$y(n) = b_0x(n) + w_1(n-1)$$
$$w_1(n) = b_1x(n) - a_1y(n) + w_2(n-1)$$
$$w_2(n) = b_2x(n) - a_2y(n)$$

All biquad filter coefficients are summarised in the ASN filter designer as:


Filter Summary
Documentation
Filter technical specification and summary.
Biquad coefficients
section gain.
forward path coefficients.
feedback path coefficients.

```
** Primary Filter (H1)**
Filter Arithmetic = Floating Point (Double Precision)
Architecture = IIR
Structure = Direct Form II Transposed
Response = Lowpass
Method = Elliptic
Biquad = Yes
Stable = Yes
Sampling Frequency = 500Hz
Filter Order = 2

Band  Frequencies (Hz)    Att/Ripple (dB)
1     0, 25               0.001
2     180.74, 250         -19.567

Biquad #1
Gain = 0.306698
B = [ 1.00000000000,  1.54712277573,  1.00000000000]
A = [ 1.00000000000, -0.11670051447,  0.20472156970]
```

Figure 2 - ASN filter designer: filter summary.

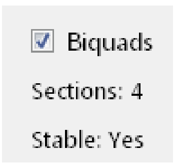
Notice that the filter summary provides the designer with a detailed overview of the designed filter, including a detailed summary of the technical specifications and the filter coefficients. The filter coefficients may be exported to third party applications for integration or further analysis, as discussed in section 1.2.

The following piece of Matlab code provides a complete implementation of the filtering operation for a Direct Form II structure biquad:

```
b=[1.0000, 1.5471,1.0000]; % feedforward coeffs
a=[1.0000, -0.1167, 0.2047]; % feedback coeffs
G=0.3067; % section gain

bg=G*b; % pre-multiply feedforward coeffs with gain

N=100;  % 100 test samples
x=randn(N,1); % generate random samples
d=zeros(3,1); % clear delay line
y=zeros(N,1); % allocate output array

for n=1:N
    d=[0;d(1:end-1)];  % update delay line
    d(1) = x(n)- a(2)*d(2)- a(3)*d(3); % compute feedback
    y(n) = bg(1)*d(1)+bg(2)*d(2)+bg(3)*d(3);  % compute feedforward
end
```

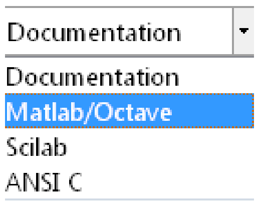
Figure 3 - Matlab code for a Direct Form II biquad filter.

The ASN filter supports the design and implementation of both single section and biquad IIR filters. For any readers wishing to experiment with single section IIR filters (a subject area not covered in this application note), unchecking the `Biquads` checkbox in the IIR filter designer will automatically apply the single section settings. Likewise, checking the `Biquads` checkbox will apply the biquad settings.


☑ Biquads
Sections: 4
Stable: Yes

1.2. Exporting coefficients to Matlab, Scilab or ANSI C

The ASN filter designer fully supports the export of filter coefficients to Matlab, Scilab or ANSI C allowing the designer to undertake further analysis or integration with a third party algorithm.



The `Documentation` view is the standard view, which is suitable for producing technical documentation. The `ANSI C` option 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 applications, such as LabView, Excel etc. Import scripts for both Matlab and Scilab are shipped with the designer in order to expedite further analysis in those domains. Frameworks for Matlab, Scilab and ANSI C (including examples) can be found in the product's installation directory.

Directory name	**Description**
`\Matlab`	Matlab filter framework.
`\Scilab`	Scilab filter framework.
`\ANSIC`	ANSI C filter framework.

1.3. Fixed point implementation

As aforementioned the `Direct Form I` filter structure is the best choice for fixed point implementation. However, before implementing the difference equation on a fixed point processor, several important data scaling considerations must be taken into account. The following discussion relates to an implementation on a 16-bit word architecture, where the theory is equally valid for any other type of word length.

1.3.1. Quantisation

In order to correctly represent the coefficients and input/output numbers, the system word length (16-bit for the purposes of this application note) is first split up into its *number of integers* and *fractional* components. The general format is given by:

Q *Num of Integers.Fraction*

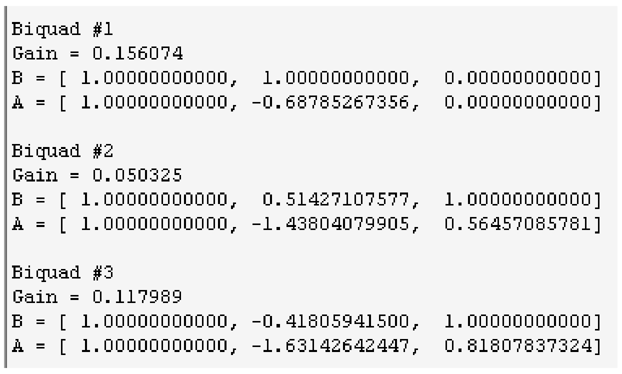
If we assume that all of data values lie within a maximum/minimum range of ±2, we can use Q1.14 format to represent all of the numbers respectively. Notice that Q1.14 format represents a maximum of $2 - 2^{-14} = 1.9999 = 0x7FFF$ and a minimum of $-2 = 0x8000$ (two's complement format).

```
WordLength = 16;
NumFractionalBits = 14;

Q=2^NumFractionalBits;
IntegerLength=2^(WordLength-NumFractionalBits-1);
MaxValue=IntegerLength-(1/Q);
MinValue=-IntegerLength;
```

Figure 4 - Quantisation Matlab code.

1.3.2. Understanding the filter summary

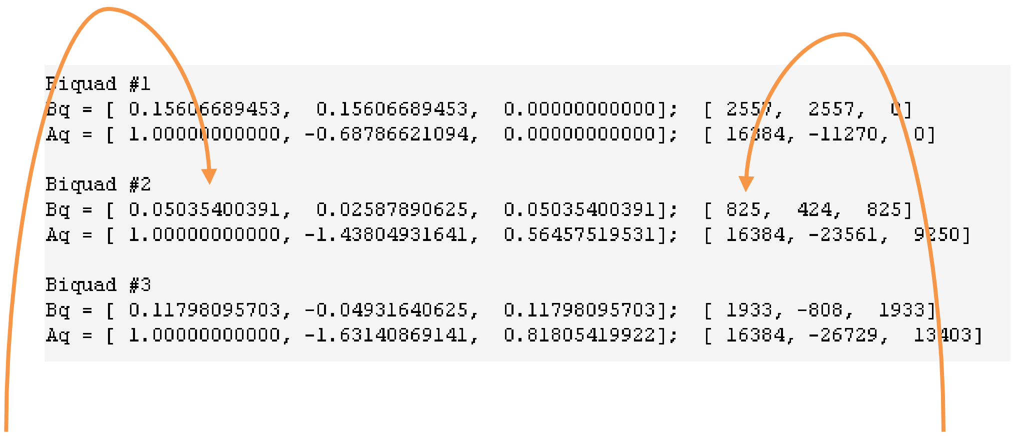
In order to fully understand the information presented in the ASN filter designer, the following example illustrates the filter coefficients obtained with double precision and with Q1.14 quantisation.



```
Biquad #1
Gain = 0.156074
B = [ 1.00000000000,  1.00000000000,  0.00000000000]
A = [ 1.00000000000, -0.68785267356,  0.00000000000]

Biquad #2
Gain = 0.050325
B = [ 1.00000000000,  0.51427107577,  1.00000000000]
A = [ 1.00000000000, -1.43804079905,  0.56457085781]

Biquad #3
Gain = 0.117989
B = [ 1.00000000000, -0.41805941500,  1.00000000000]
A = [ 1.00000000000, -1.63142642447,  0.81807837324]
```

double precision

Applying Q1.14 (note the effects of quantisation on the coefficient values):



```
Biquad #1
Bq = [ 0.15606689453,  0.15606689453,  0.00000000000];  [ 2557,  2557,   0]
Aq = [ 1.00000000000, -0.68786621094,  0.00000000000];  [ 16384, -11270,   0]

Biquad #2
Bq = [ 0.05035400391,  0.02587890625,  0.05035400391];  [ 825,   424,   825]
Aq = [ 1.00000000000, -1.43804931641,  0.56457519531];  [ 16384, -23561,   9250]

Biquad #3
Bq = [ 0.11798095703, -0.04931640625,  0.11798095703];  [ 1933, -808,   1933]
Aq = [ 1.00000000000, -1.63140869141,  0.81805419922];  [ 16384, -26729,   13403]
```

floating point equivalent:

$$\frac{825}{2^{14}} = 0.050354$$

decimal coefficient value:

$$\text{ceil}\,(2^{14} \times 0.050325) = 825$$

The `Cascade Scaling Factors` parameters presented in the summary are only used for the Direct Form II structure (see section 1.3.3.1), and can therefore be ignored for a Direct Form I implementation scheme.

1.3.3. Direct Form II scaling

In order to ensure that the feedback path, *w(n)* will not overflow, 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 LInf 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.
LInf norm	$G = \max \lvert W(w) \rvert$	LInf norm ensure that the filter will not overflow when a sine wave is applied.

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

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

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

1.3.3.1. Biquad cascade scaling

A biquad cascade comprised of three biquads is shown below.

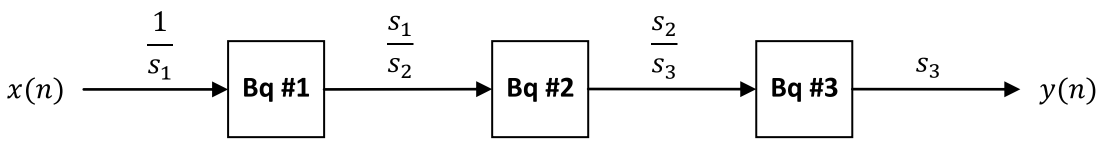


Figure 5 - biquad cascade scaling concept.

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.

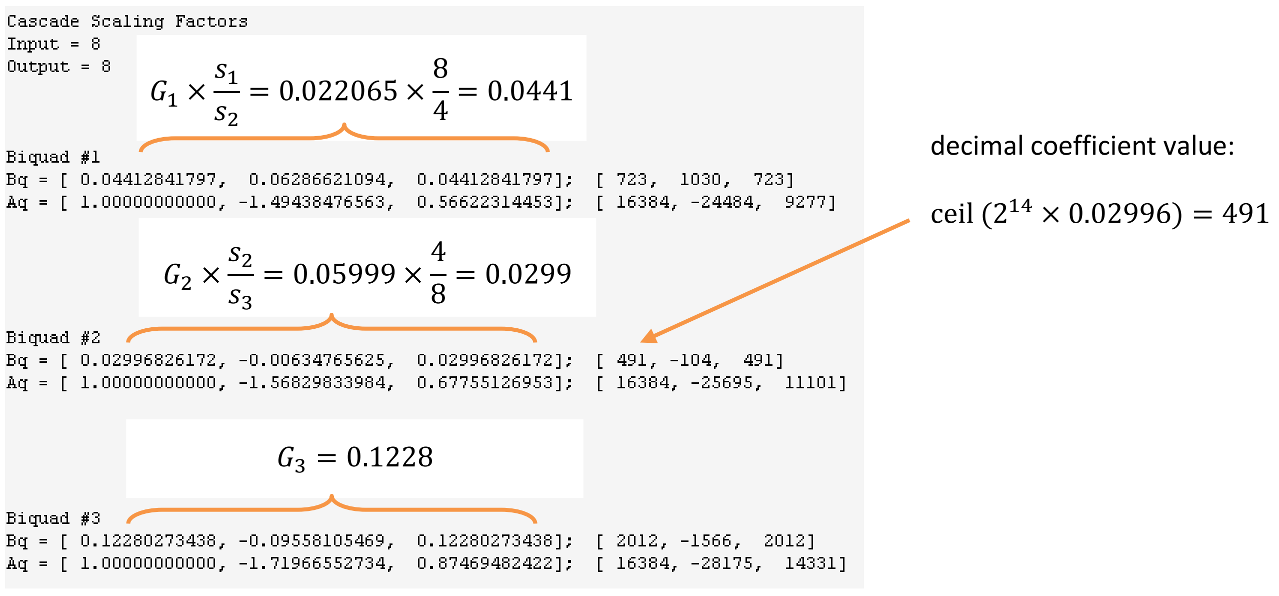
1.3.3.2. Example

In order to fully understand the information presented in the ASN filter designer with scaling, 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
```

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

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

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

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

$$G_3 = 0.1228$$

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

decimal coefficient value:

$$\text{ceil}\,(2^{14} \times 0.02996) = 491$$

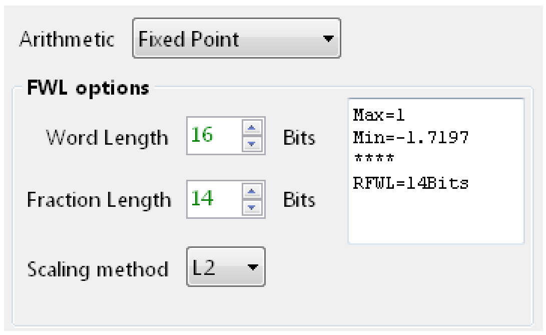
The decimal coefficients may be directly inserted into a fixed point algorithm for implementation.

1.3.4. Comparing spectra obtained by different arithmetic rules

In order to improve clarity and overall computation speed, the ASN filter designer only displays spectra (i.e. magnitude, phase etc.) based on the current arithmetic rules. This is somewhat different to other tools that display multi-spectra obtained by (for example) fixed point and double precision arithmetic. For any users wishing to compare spectra you may simply switch between arithmetic settings by changing the `Arithmetic` method (see section 1.3.5). The designer will then automatically re-compute the filter coefficients using the selected arithmetic rules and the current technical specification. The chart will then be updated using the current zoom settings.

1.3.5. Analytics

The ASN filter designer assists designers with extra analytics in order to help choose a suitable finite word length (FWL). The recommend FWL (RFWL) value is based on the analysis of the unquantised filter coefficients values.



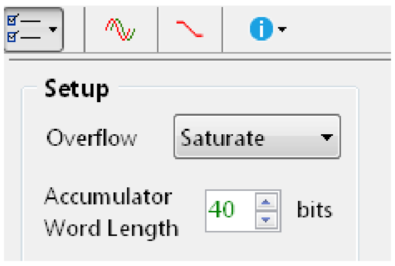
Arithmetic: Fixed Point
Word Length: 16 Bits
Fraction Length: 14 Bits
Scaling method: L2

```
Max=1
Min=-1.7197
****
RFWL=14Bits
```

For the example considered herein, it can be seen that the tool summarises the maximum and minimum coefficient values obtained with double precision. As seen, a recommendation of `14Bits` is given as the largest negative value is `-1.7197` which with a word length of `16Bits` would require at least a fraction length of `14Bits`.

1.3.6. Validating the design with the signal analyser

A design may be validated with the signal analyser, where both time and frequency domain plots are supported. A comprehensive signal generator is fully integrated into the signal analyser allowing designers to test their filters with a variety of input signals, such as sine waves, white noise or even external test data. Please refer to the ASN filter designer user's guide (ASN15-DOC001) for more information.



Overflow: Saturate
Accumulator Word Length: 40 bits

For fixed point implementations the tool allows designers to specify the overflow arithmetic rules as: `Saturate` or `Wrap`. Also, the Accumulator Word Length may be set between `16-40Bits` allowing designers to quickly find the optimum settings to suit their application.

1.4. Technical references and product resources

This application note 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.

- ASN Filter Designer product home page: http://www.advsolned.com/asn_filter_designer.html
- ASN Technical support: support@advsolned.com

Document Revision Status

Rev.	Description	Date
1	Document released for v1.0.	10/01/2012
2	Document updated for v2.0	28/09/2015
3	Document updated for v3.0	10/02/2016