Linear-Phase Least-Squares FIR Filter Synthesis

May 20, 2017
William E. Jones
http://www.wejc.com

1 Description

This document describes a software tool that synthesizes linear-phase FIR filters. This tool is presented in the form of a C++ class named *LS_FIR_Filter*.

2 Filter Segment Types

Frequencies are expressed in fractions of the Nyquist rate in these algorithms and this software.

$$0 \le F < 1 \tag{1}$$

Amplitude can be any real-valued number, including zero, if absolute weighting is used. If relative error weighting is used then amplitudes must be greater than zero as these amplitudes are specified by their logarithms.

2.1 Linear Segments

The amplitude response of a *linear segment* is defined by

$$R(f) = Bf + A \tag{2}$$

We define the line segment by its endpoints so

$$freq:[F_0, F_1] \qquad amp:[A_0, A_1] \tag{3}$$

From this

$$B = \frac{(A_1 - A_0)}{(F_1 - F_o)} \tag{4}$$

$$A = A_0 - B F_o \tag{5}$$

2.2 Exponential Segments

The amplitude response of an *exponential segment* is defined by

$$R(f) = e^{Bf+A} \tag{6}$$

We define the line segment by its endpoints so

$$freq:[F_0, F_1] \qquad amp:[\log(A_0), \log(A_1)] \tag{7}$$

From this

$$B = \frac{(\log(A_1) - \log(A_0))}{(F_1 - F_o)} \tag{8}$$

$$A = \log(A_0) - B F_0 \tag{9}$$

3 Error Types

The error type determines how the amplitude square error between the filter prototype and the filter realization weighted.

3.1 Absolute Error

Absolute weighting is the basic form where the sum of the square errors is minimized.

3.2 Relative Error

Relative error weighting minimizes the relative square error rather than the absolute square error. The error is scaled by the magnitude of the filter response so tightens-up the response at the lower amplitude values. Compare the filter response in Section 4.1 below with that of Section 4.5 below. Note that the filter representation at lower amplitudes is better with the relative weighting scheme.

4 Diagnostic Runs

The following sections preset the diagnostic runs that the software uses to test for its proper operation.

4.1 1023-coefficient exponential segment absolute weighted

A 1023-coefficient, exponential segment, absolutely weighted, FIR filter is generated by

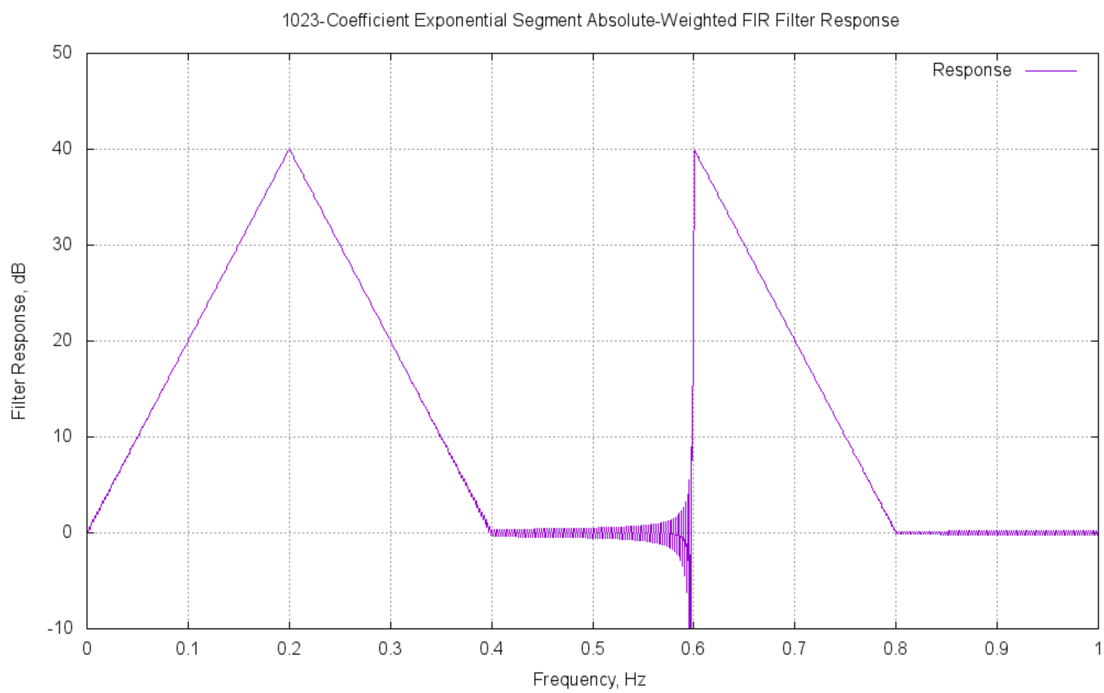
```
LS_FIR_Filter       LSF;

LSF.Add( LS_FIR_Filter::eExp_AbsErr,  0, .2,    1, 100 );
LSF.Add( LS_FIR_Filter::eExp_AbsErr, .2, .4,  100,   1 );
LSF.Add( LS_FIR_Filter::eExp_AbsErr, .4, .6,    1,   1 );
LSF.Add( LS_FIR_Filter::eExp_AbsErr, .6, .8,  100,  1 );
LSF.Add( LS_FIR_Filter::eExp_AbsErr, .8,  1,    1,   1 );

const unsigned int              Filter_Len = 1023;
vector<complex<double>>     Filter(Filter_Len), In(FFT_Len,0), Out(FFT_Len);

LSF.GenFilter( Filter_Len, Filter );
```

The phase response is linear and the realized filter's frequency response is given by



4.2 1024-coefficient exponential segment absolute weighted

A 1024-coefficient, exponential segment, absolutely weighted, FIR filter is synthesized by

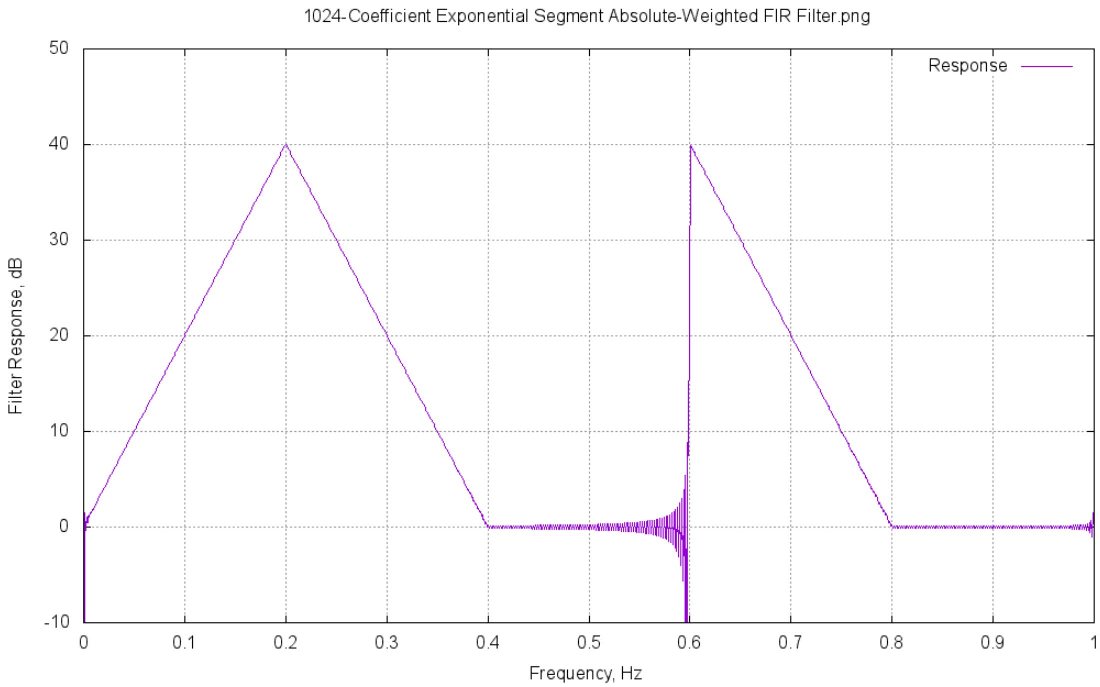
```
LS_FIR_Filter       LSF;

LSF.Add( LS_FIR_Filter::eExp_AbsErr,  0, .2,    1, 100 );
LSF.Add( LS_FIR_Filter::eExp_AbsErr, .2, .4,  100,   1 );
LSF.Add( LS_FIR_Filter::eExp_AbsErr, .4, .6,    1,   1 );
LSF.Add( LS_FIR_Filter::eExp_AbsErr, .6, .8,  100,   1 );
LSF.Add( LS_FIR_Filter::eExp_AbsErr, .8,  1,    1,   1 );

const unsigned int              Filter_Len = 1024;
vector<complex<double>>     Filter(Filter_Len), In(FFT_Len,0), Out(FFT_Len);

LSF.GenFilter( Filter_Len, Filter );
```

The phase response is linear and the realized filter's frequency response is given by



4.3 1023-coefficient linear segment absolute weighted

A 1023-coefficient, linear segment, absolutely weighted, FIR filter is synthesized by

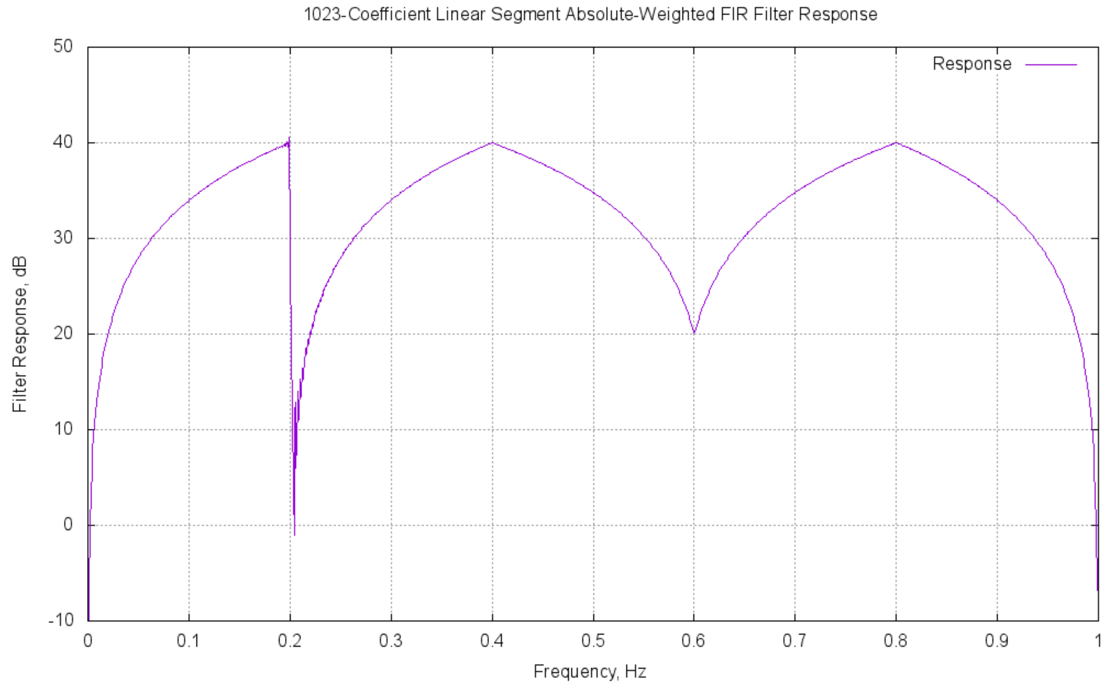
```
LS_FIR_Filter       LSF;

LSF.Add( LS_FIR_Filter::eLin_AbsErr,  0, .2,   0, 100 );
LSF.Add( LS_FIR_Filter::eLin_AbsErr, .2, .4,   0, 100 );
LSF.Add( LS_FIR_Filter::eLin_AbsErr, .4, .6, 100,  10 );
LSF.Add( LS_FIR_Filter::eLin_AbsErr, .6, .8,  10, 100 );
LSF.Add( LS_FIR_Filter::eLin_AbsErr, .8,  1, 100,   0 );

const unsigned int              Filter_Len = 1023;
vector<complex<double>>     Filter(Filter_Len), In(FFT_Len,0), Out(FFT_Len);

LSF.GenFilter( Filter_Len, Filter );
```

The phase response is linear and the realized filter's frequency response is given by



4.4 1024-coefficient linear segment absolute weighted

A 1024-coefficient, linear segment, absolutely weighted, FIR filter is synthesized by

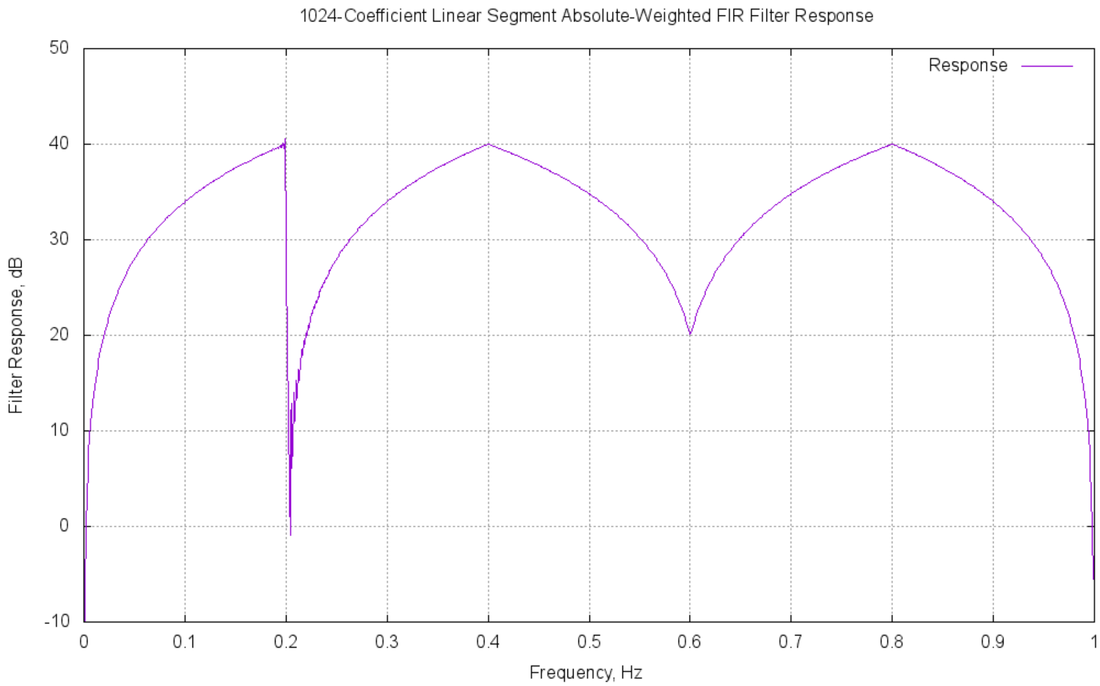
```
LS_FIR_Filter       LSF;

LSF.Add( LS_FIR_Filter::eLin_AbsErr,  0, .2,   0, 100 );
LSF.Add( LS_FIR_Filter::eLin_AbsErr, .2, .4,   0, 100 );
LSF.Add( LS_FIR_Filter::eLin_AbsErr, .4, .6, 100,  10 );
LSF.Add( LS_FIR_Filter::eLin_AbsErr, .6, .8,  10, 100 );
LSF.Add( LS_FIR_Filter::eLin_AbsErr, .8,  1, 100,   0 );

const unsigned int              Filter_Len = 1024;
vector<complex<double>>         Filter(Filter_Len), In(FFT_Len,0), Out(FFT_Len);

LSF.GenFilter( Filter_Len, Filter );
```

The phase response is linear and the realized filter's frequency response is given by



4.5 1023-coefficient exponential segment relative weighted

A 1023-coefficient, exponential segment, relatively weighted, FIR filter is synthesized by

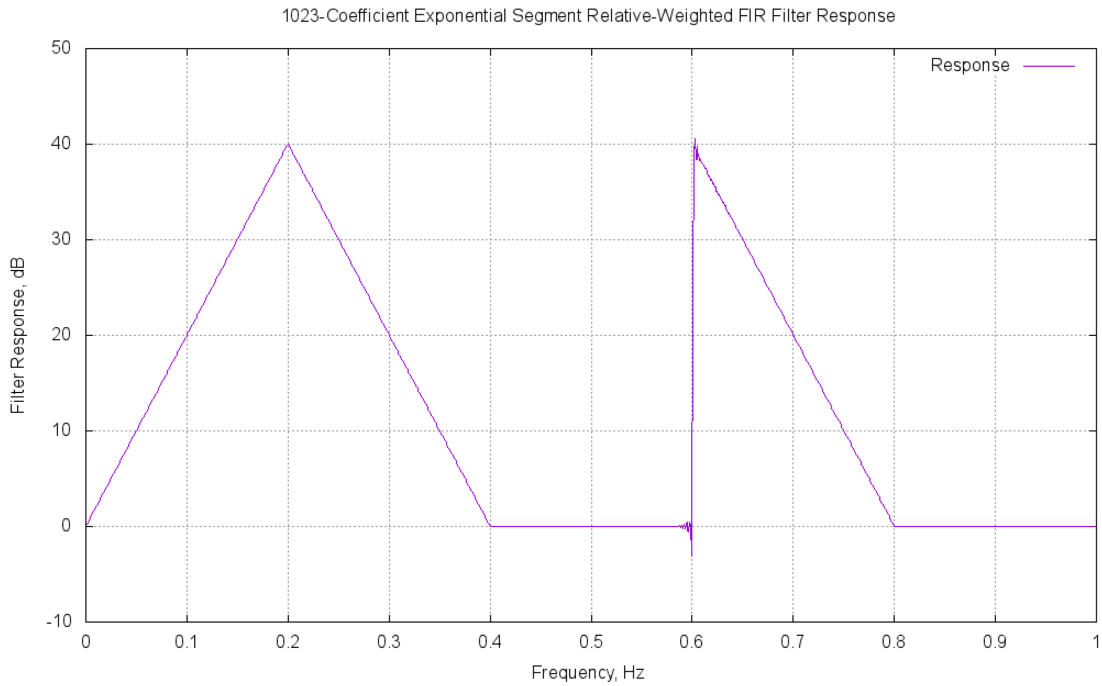
```
LS_FIR_Filter       LSF;

LSF.Add( LS_FIR_Filter::eExp_RelErr,  0, .2,     1, 100 );
LSF.Add( LS_FIR_Filter::eExp_RelErr, .2, .4,   100,   1 );
LSF.Add( LS_FIR_Filter::eExp_RelErr, .4, .6,     1,   1 );
LSF.Add( LS_FIR_Filter::eExp_RelErr, .6, .8,   100,   1 );
LSF.Add( LS_FIR_Filter::eExp_RelErr, .8,  1,     1,   1 );

const unsigned int              Filter_Len = 1023;
vector<complex<double>>         Filter(Filter_Len), In(FFT_Len,0), Out(FFT_Len);

LSF.GenFilter( Filter_Len, Filter );
```

The phase response is linear and the realized filter's frequency response is given by



4.6 1024-coefficient exponential segment relative weighted

A 1024-coefficient, exponential segment, relatively weighted, FIR filter is synthesized by

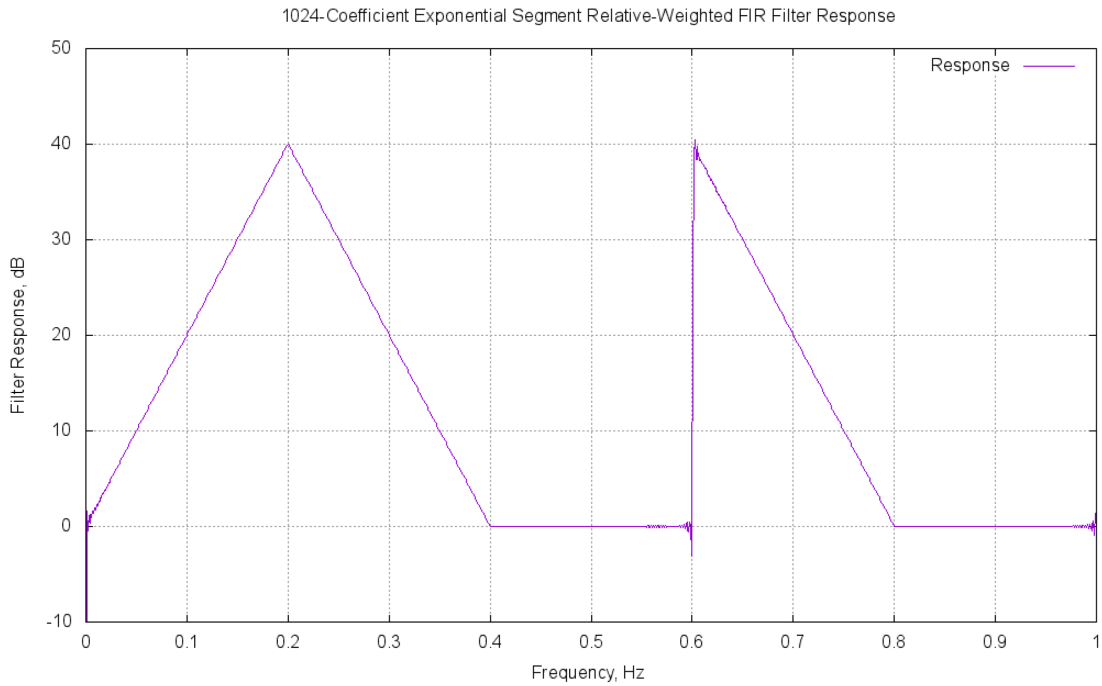
```
LS_FIR_Filter      LSF;

LSF.Add( LS_FIR_Filter::eExp_RelErr,  0, .2,    1, 100 );
LSF.Add( LS_FIR_Filter::eExp_RelErr, .2, .4,  100,   1 );
LSF.Add( LS_FIR_Filter::eExp_RelErr, .4, .6,    1,   1 );
LSF.Add( LS_FIR_Filter::eExp_RelErr, .6, .8,  100,   1 );
LSF.Add( LS_FIR_Filter::eExp_RelErr, .8,  1,    1,   1 );

const unsigned int              Filter_Len = 1024;
vector<complex<double>>      Filter(Filter_Len), In(FFT_Len,0), Out(FFT_Len);

LSF.GenFilter( Filter_Len, Filter );
```

The phase response is linear and the realized filter's frequency response is given by



4.7 1023-coefficient linear segment relative weighted

A 1023-coefficient, linear segment, relatively weighted, FIR filter is synthesized by

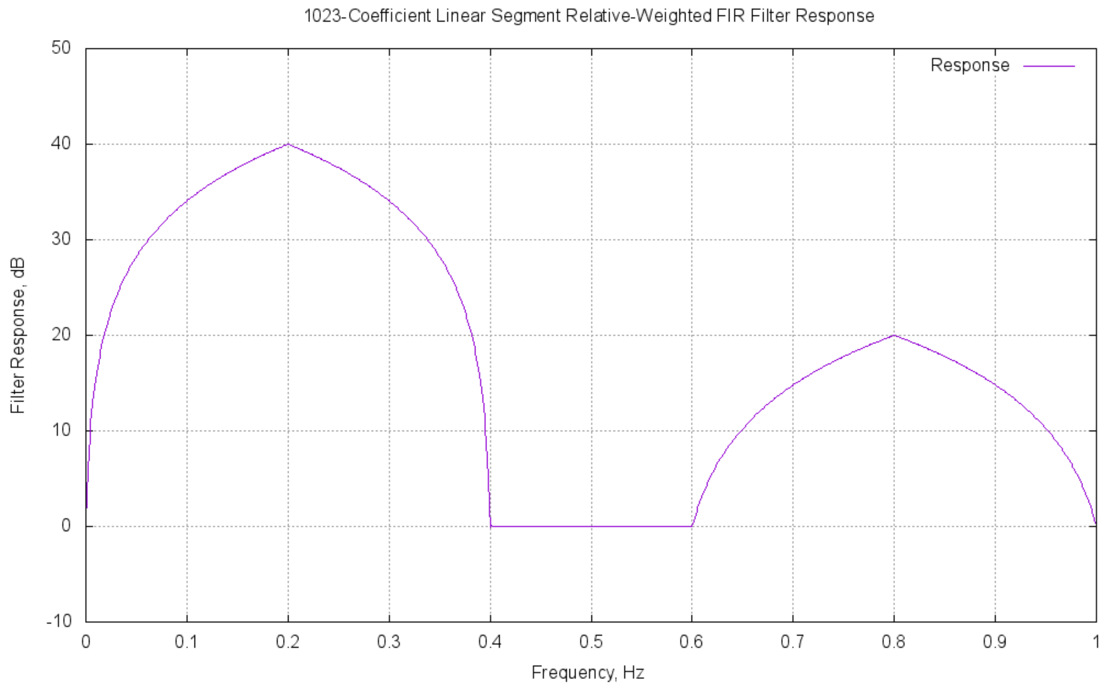
```
LS_FIR_Filter       LSF;

LSF.Add( LS_FIR_Filter::eLin_RelErr,  0, .2,    1, 100 );
LSF.Add( LS_FIR_Filter::eLin_RelErr, .2, .4,   100,  1 );
LSF.Add( LS_FIR_Filter::eLin_RelErr, .4, .6,    1,   1 );
LSF.Add( LS_FIR_Filter::eLin_RelErr, .6, .8,    1,  10 );
LSF.Add( LS_FIR_Filter::eLin_RelErr, .8,  1,    10,  1 );

const unsigned int              Filter_Len = 1023;
vector<complex<double>>      Filter(Filter_Len), In(FFT_Len,0), Out(FFT_Len);

LSF.GenFilter( Filter_Len, Filter );
```

The phase response is linear and the realized filter's frequency response is given by



4.8 1024-coefficient linear segment relative weighted

A 1024-coefficient, linear segment, relatively weighted, FIR filter is synthesized by

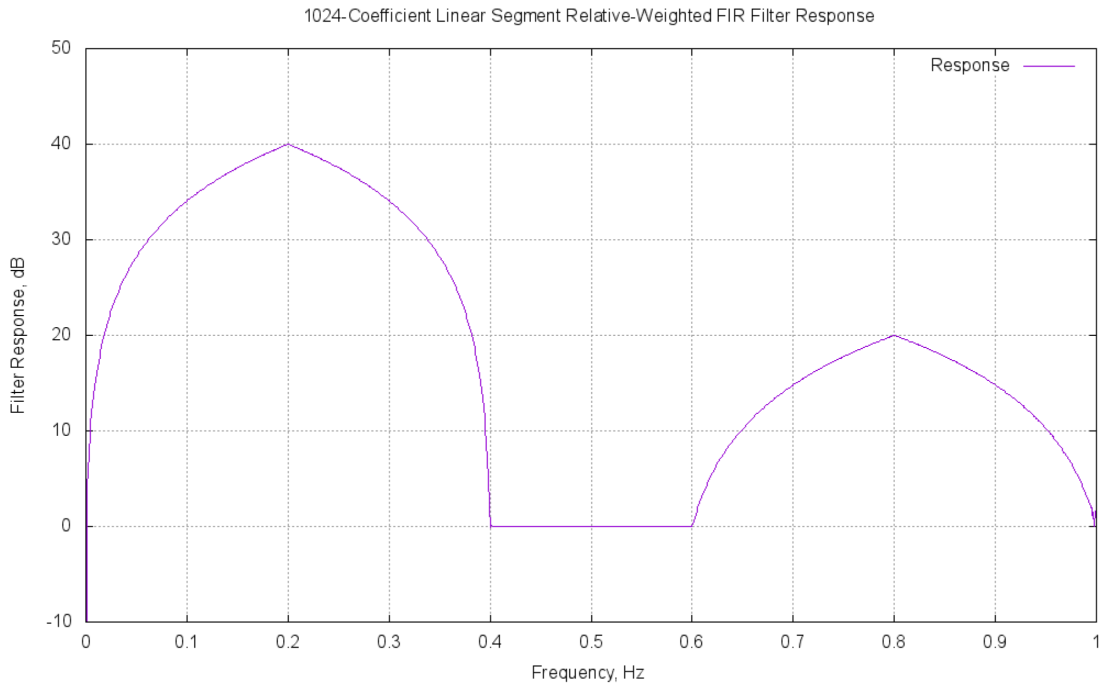
```
LS_FIR_Filter       LSF;

LSF.Add( LS_FIR_Filter::eLin_RelErr,  0, .2,    1, 100 );
LSF.Add( LS_FIR_Filter::eLin_RelErr, .2, .4,   100,  1 );
LSF.Add( LS_FIR_Filter::eLin_RelErr, .4, .6,    1,   1 );
LSF.Add( LS_FIR_Filter::eLin_RelErr, .6, .8,    1,  10 );
LSF.Add( LS_FIR_Filter::eLin_RelErr, .8,  1,    10,  1 );

const unsigned int              Filter_Len = 1024;
vector<complex<double>>      Filter(Filter_Len), In(FFT_Len,0), Out(FFT_Len);

LSF.GenFilter( Filter_Len, Filter );
```

The phase response is linear and the realized filter's frequency response is given by



5 References

[1] A.G. Jaffer, W.E. Jones, "Weighted least-squares design and characterization of complex FIR filters", Signal Processing IEEE Transactions on, vol. 43, pp. 2398-2401, 1995, ISSN 1053-587X.

[2] A.G. Jaffer, W.E. Jones, "Constrained Least-Squares Design and Characterization of Affine Phase Complex FIR Filters", Presented at the 27th Annual Asilomar Conference, Nov. 1-3,1993, Pacific Grove California.

[3] A.G. Jaffer, W.E. Jones, T.J. Abatzoglou, "Weighted least-squares design of linear-phase and arbitrary 2-D complex FIR filters", Acoustics Speech and Signal Processing 1995. ICASSP-95. 1995 International Conference on, vol. 2, pp. 1256-1259 vol.2, 1995, ISSN 1520-6149.