Digital filter design

In RFC-A or RFC-S mode it is possible to replace the low pass filter (defined with *Current Reference Filter 1 Time Constant* (04.012) or *Low-pass Filter Cut-off Frequency* (04.050), etc.) and/or the notch filter (defined with *Notch Centre Frequency* (04.031) and *Notch Filter Bandwidth* (04.032) which are applied to the torque reference with user customisable filters. These user customisable filters are based on the bi-quad format which can be used to synthesise first and second order filters. This document describes the process required to design a digital filter and calculate the user parameters to set up the bi-quad filters in the drive.

If any of the parameters *Bi-quad Filter1: a0* (04.051) to *Bi-quad Filter1: b2* (04.055) are changed from their default value of 0.0000 the on-board low pass filter is disabled and bi-quad filter 1 is used instead.

If any of the parameters *Bi-quad Filter2: a0* (04.056) to *Bi-quad Filter2: b2* (04.060) are changed from their default value of 0.0000 the on-board notch filter is disabled and bi-quad filter 2 is used instead.

It is possible to use one bi-quad filter in combination with either the on-board low-pass or on-board notch filter by configuring the appropriate bi-quad filter.

A low-pass filter is used as an example to explain the process using bi-quad filter 1 in the drive. Examples are also given for a notch filter and a lag-lead filter.

Process to design a low-pass filter

A bi-quad filter has the following transfer function in the s domain.

$$H(s) = \frac{As^2 + Bs + C}{Ds^2 + Es + F}$$

To use this for a particular type of filter the coefficients must be modified to represent those of the required filter. For example, a low-pass with a cut-off frequency of f_{co} Hz is given by the following transfer function in the s domain.

$$H(s) = \frac{1}{(1/2\pi f_{co})s + 1} = \frac{1}{(1/\omega_{co})s + 1} = \frac{\omega_{co}}{s + \omega_{co}}$$

$$\omega_{co} = 2\pi f_{co}$$

$$A = 0, B = 0, C = \omega_{co}, D = 0, E = 1, F = \omega_{co}$$

This transfer function must first be converted into the z domain. However, this process causes a distortion of the filter characteristic and the cut-off frequency may no longer be accurate especially if the cut-off frequency is close to the Nyquist frequency. The sample frequency for the filter is 4kHz, and so the Nyquist frequency is 2kHz. Even if the following correction process is followed it is best for the characteristic frequencies of the filter to be less than 50% of the Nyquist frequency. Therefore, the cut-off frequency for the low-pass filter is best kept below 1kHz. If the correction process is followed it is possible to exceed this level, but the filter characteristic may start to deviate significantly from the required characteristic if the characteristic frequencies are higher than 60% of the Nyquist frequency. To compensate for the frequency distortion the characteristic frequencies of the filter must be modified using the following equation. The required characteristic frequency ω_{char}* must be converted to ω_{char} which is the modified frequency that must be used in the filter design.

$$\omega_{char} = \frac{2}{T}\, tan(\omega_{char}{}^{*}T/2)$$

Where T is the sample time 1/4000Hz = $250\text{x}10^{-6}$.

For the example low-pass filter, the required cut-off frequency of ω_{co}* must be converted to ω_{co}.

$$\omega_{co} = \frac{2}{T}\, tan(\omega_{co}{}^{*}T/2)$$

If the required cut-off frequency is 500Hz then ω_{co} is given by

$$\omega_{co} = \frac{2}{250 \times 10^{-6}}\, tan(2\pi \times 500Hz\ \times 250 \times 10^{-6}/2) = 3314rad/s$$

In this case the cut-off frequency is not close to the Nyquist frequency, and so the change from 2πx500=3142rad/s is small. If the cut-off frequency is higher the change is larger.

The bilinear transform is used to convert from the s domain to the z domain giving an equation that can be used to give the digital filter coefficients. To use the bilinear transform the following substitution is used.

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$$

For the low-pass filter

$$H(s) = \frac{\omega_{co}}{s + \omega_{co}}$$

and so

$$H(z) = \frac{\omega_{co}}{\left(\frac{2}{T}\left(\frac{z-1}{z+1}\right)\right) + \omega_{co}}$$

This needs to be rearranged into the following format so that it can be used to create a difference equation.

$$H(z) = \frac{a_0 z + a_1}{z + b_1}$$

For the low-pass filter

$$H(z) = \frac{(\omega_{co}T/2)(z+1)}{(z-1) + (\omega_{co}T/2)(z+1)}$$

$$H(z) = \frac{(\omega_{co}T/2)z + (\omega_{co}T/2)}{z\big((\omega_{co}T/2)+1\big) + (\omega_{co}T/2) - 1}$$

$$H(z) = \frac{\frac{\omega_{co}T/2}{(\omega_{co}T/2)+1}z + \frac{\omega_{co}T/2}{(\omega_{co}T/2)+1}}{z + \frac{(\omega_{co}T/2)-1}{(\omega_{co}T/2)+1}}$$

$$H(z) = \frac{\frac{1}{1+(2/\omega_{co}T)}z + \frac{1}{1+(2/\omega_{co}T)}}{z + \frac{1-(2/\omega_{co}T)}{1+(2/\omega_{co}T)}}$$

And so

$$a_0 = a_1 = \frac{1}{1 + 2/(\omega_{co}T)}$$

$$b_1 = \frac{1 - 2/(\omega_{co}T)}{1 + 2/(\omega_{co}T)}$$

The transfer function in the z domain can be rearranged.

$$H(z) = \frac{O(z)}{I(z)} = \frac{a_0 z + a_1}{z + b_1} = \frac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1}}$$

$O(z)(1 + b_1 z^{-1}) = I(z)(a_0 + a_1 z^{-1})$

This can then be converted to a difference equation to implement the filter.

$O_N = a_0 I_N + a_1 I_{N-1} - b_1 O_{N-1}$

This last stage is only shown to explain how the coefficients are used. The values entered in the drive parameters are a_0, a_1 and b_1. The drive parameters for a_2 and b_2 are left at their default values of 0.0000. For the example low-pass filter with a cut-off frequency of 500Hz the value of ω_{co} which has been modified to correct for the filter distortion is 3314rad/s.

$$a_0 = a_1 = \frac{1}{1 + 2/(3314 \times 250 \times 10^{-6})} = 0.2929$$

$$b_1 = \frac{1 - 2/(\omega_{co}T)}{1 + 2/(\omega_{co}T)} = \frac{1 - 2/(3314 \times 250 \times 10^{-6})}{1 + 2/(3314 \times 250 \times 10^{-6})} = -0.4142$$

The parameter set-up for this filter is

Bi-quad Filter1: a0 (04.051) = 0.2929
Bi-quad Filter1: a1 (04.052) = 0.2929
Bi-quad Filter1: a2 (04.053) = 0.0000
Bi-quad Filter1: b1 (04.054) = -0.4142
Bi-quad Filter1: b2 (04.055) = 0.0000

Confirming the low-pass filter response

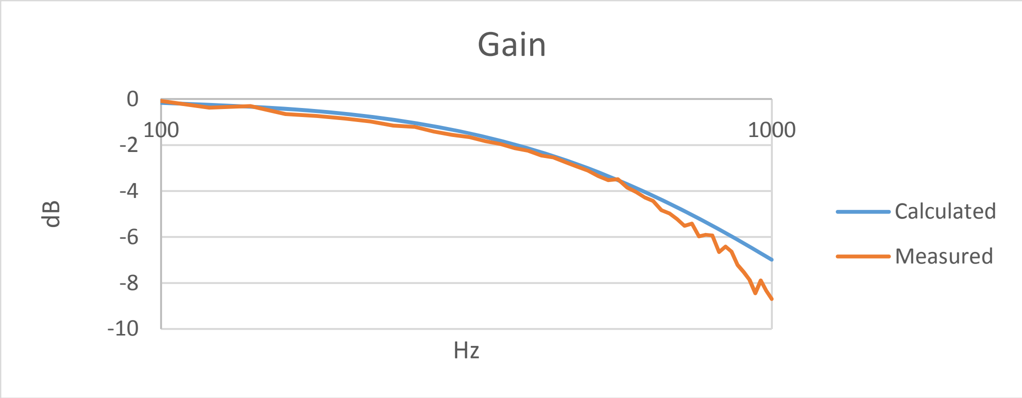
It is possible to confirm the filter response using the CT Analyser Control Response application tab and a Unidrive M7xx drive or Digitax HD M7xx drive. The following procedure should be used.

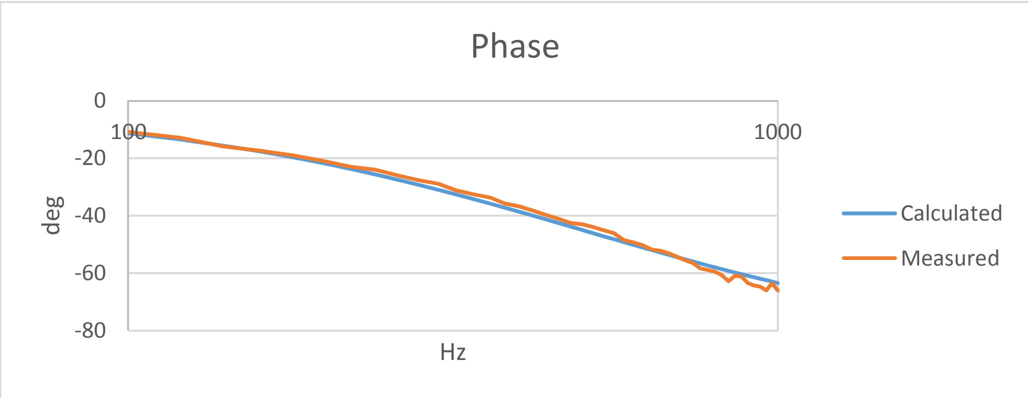
1. To avoid unexpected results due to resonances it is best to use a servo motor with no load connected to the shaft.
2. To obtain the best results an EnDat 2.2 device should be used (*P1 Device Type* (03.038)="Endat") with 25 bit resolution per turn.
3. No filters should be enabled, i.e. *Current Reference Filter 1 Time Constant* (04.012), Notch Filter Bandwidth (04.032), *Low-pass Filter Cut-off Frequency* (04.050), *Bi-quad Filter1: a0* (04.051) to *Bi-quad Filter1: b2* (04.055), and *Bi-quad Filter2: a0* (04.056) to *Bi-quad Filter2: b2* (04.060) should all be zero.
4. A stationary auto-tune should be carried out to set up the drive for the motor.
5. An inertia auto-tune (either Inertia1 or Inertia2) should be carried out to measure the motor inertia.
6. Speed Controller Set-up Method (03.017) should be set to "Bandwidth" and the *Bandwidth* (03.020) should be increased from its default value to give good performance. Normally it will be possible to increase the bandwidth to at least 100Hz with an EnDat encoder. However, it is important that there is no system resonance or position feedback noise, and so if the motor is acoustically noisy when running the bandwidth must be reduced.
7. The motor should be made to run at a moderate speed, i.e. 25% of rated speed.
8. The forward-loop characteristic should be measured over the required frequency range using CT Analyser and the results saved as an Excel file.
9. The required filter should be set up in parameters *Bi-quad Filter1: a0* (04.051) to *Bi-quad Filter1: b2* (04.055) and the forward-loop characteristic should be measured and saved.

The gain and phase of the filter at each test frequency is given by

Filter gain = No filter measured gain – Gain measured with filter present
Filter phase = No filter measured phase – Phase measured with filter present

The results given below show the measured results compared with the calculated characteristic for an ideal low-pass filter with a cut-off frequency of 500Hz. These have been calculated and plotted in Excel.





Notch filter design

A notch filter in the s domain is given by

$$H(s) = \frac{s^2 + 2G\xi\omega_0 s + {\omega_0}^2}{s^2 + 2\xi\omega_0 s + {\omega_0}^2}$$

ω_0 is the centre frequency in rad/s.

ξ is the damping factor. The Q of the filter and the bandwidth (frequency between the 3dB points in rad/s, which are not symmetrical about the centre frequency) are related as follows.

$$2\xi = 1/Q = \omega_{bw}/\omega_0$$

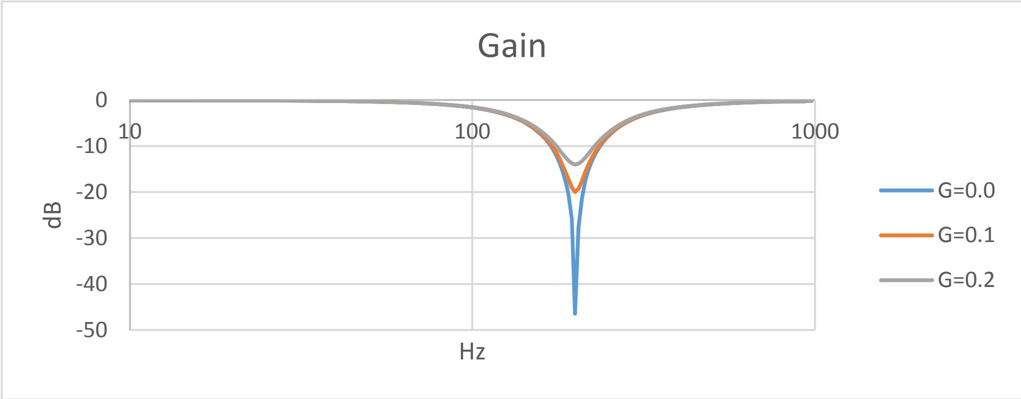
Therefore

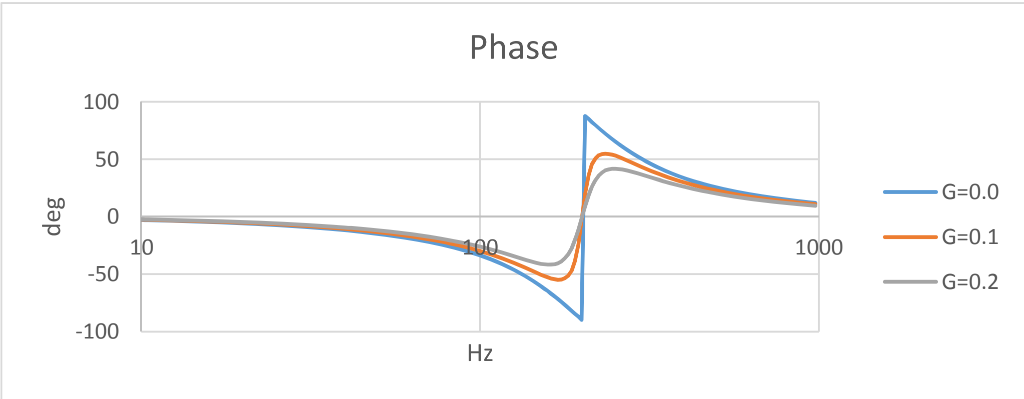
$$H(s) = \frac{s^2 + G\omega_{bw} s + {\omega_0}^2}{s^2 + \omega_{bw} s + {\omega_0}^2}$$

G is the gain at the centre of the notch. For example, if G=0.1 the gain of the filter at the centre frequency is 0.1 (-20dB). If G=0 then the filter becomes a standard notch filter with a theoretical gain of zero at the centre frequency.

$$H(s) = \frac{s^2 + {\omega_0}^2}{s^2 + \omega_{bw} s + {\omega_0}^2}$$

Examples with different gain settings are shown below.





The centre frequency and bandwidth must first be modified to prevent frequency distortion. If $\omega_0{}^*$ is the required centre frequency then

$$\omega_0 = \frac{2}{T}\, tan(\omega_0{}^*\, T/2)$$

If the required bandwidth is $\omega_{bw}{}^*$ then the required lower -3dB point is $\omega_{-3dB}{}^* = \omega_0{}^* - \omega_{bw}{}^*/2$. Then

$$\omega_{-3dB} = \frac{2}{T}\, tan(\omega_{-3dB}{}^*\, T/2)$$

And so, $\omega_{bw} = (\omega_0 - \omega_{-3dB}) \times 2$.

As mentioned previously the 3dB points are not symmetrical either side of the centre frequency, and so this gives a resulting digital filter that has approximately the correct bandwidth. Normally an accurate centre frequency is important, and accurate bandwidth is less important.

Using the bilinear transform

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$$

$$H(z) = \frac{\left(\frac{2}{T}\left(\frac{z-1}{z+1}\right)\right)^2 + G\omega_{bw}\left(\frac{2}{T}\left(\frac{z-1}{z+1}\right)\right) + \omega_0{}^2}{\left(\frac{2}{T}\left(\frac{z-1}{z+1}\right)\right)^2 + \omega_{bw}\left(\frac{2}{T}\left(\frac{z-1}{z+1}\right)\right) + \omega_0{}^2}$$

$$H(z) = \frac{(z-1)^2 + G\left(\frac{\omega_{bw}T}{2}\right)(z-1)(z+1) + \left(\frac{\omega_0 T}{2}\right)^2 (z+1)^2}{(z-1)^2 + \left(\frac{\omega_{bw}T}{2}\right)(z-1)(z+1) + \left(\frac{\omega_0 T}{2}\right)^2 (z+1)^2}$$

$$H(z)=\frac{z^2-2z+1+G\left(\frac{\omega_{bw}T}{2}\right)z^2-G\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2z^2+2\left(\frac{\omega_0T}{2}\right)^2z+\left(\frac{\omega_0T}{2}\right)^2}{z^2-2z+1+\left(\frac{\omega_{bw}T}{2}\right)z^2-\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2z^2+2\left(\frac{\omega_0T}{2}\right)^2z+\left(\frac{\omega_0T}{2}\right)^2}$$

$$H(z)=\frac{z^2\left(1+G\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2\right)+z\left(-2+2\left(\frac{\omega_0T}{2}\right)^2\right)+\left(1-G\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2\right)}{z^2\left(1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2\right)+z\left(-2+2\left(\frac{\omega_0T}{2}\right)^2\right)+\left(1-\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2\right)}$$

$$H(z)=\frac{z^2\left(\frac{1+G\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}{1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}\right)+z\left(\frac{-2+2\left(\frac{\omega_0T}{2}\right)^2}{1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}\right)+\left(\frac{1-G\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}{1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}\right)}{z^2+z\left(\frac{-2+2\left(\frac{\omega_0T}{2}\right)^2}{1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}\right)+\left(\frac{1-\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}{1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}\right)}$$

$$H(z)=\frac{a_0z^2+a_1z+a_2}{z^2+b_1z+b_2}$$

And so

$$a_0=\frac{1+G\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}{1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}$$

$$a_1=\frac{2\left(\left(\frac{\omega_0T}{2}\right)^2-1\right)}{1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}$$

$$a_2=\frac{1-G\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}{1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}$$

$$b_1=a_1$$

$$b_2=\frac{1-\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}{1+\left(\frac{\omega_{bw}T}{2}\right)+\left(\frac{\omega_0T}{2}\right)^2}$$

If the required filter characteristics are f_0=500Hz, f_{bw}=400Hz, G=0.2 then the centre frequency and bandwidth must be modified to correct for frequency distortion.

$$\omega_o=\frac{2}{T}\,tan(\omega_o{}^*T/2)=\frac{2}{250\times10^{-6}}\,tan(2\pi\times500Hz\ \times250\times10^{-6}/2)=3314rad/s$$

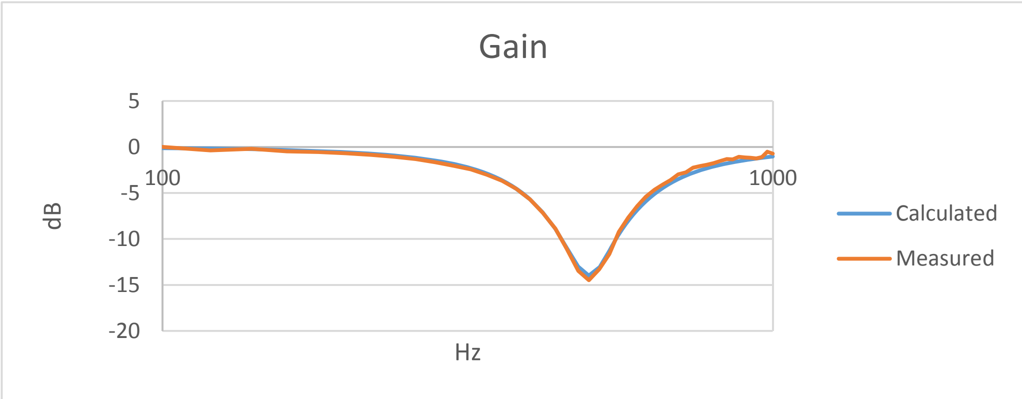
$$\omega_{-3dB}{}^*=\omega_o{}^*-\omega_{bw}{}^*/2=2\pi\times500Hz-(2\pi\times400Hz)/2=2\pi\times300Hz$$

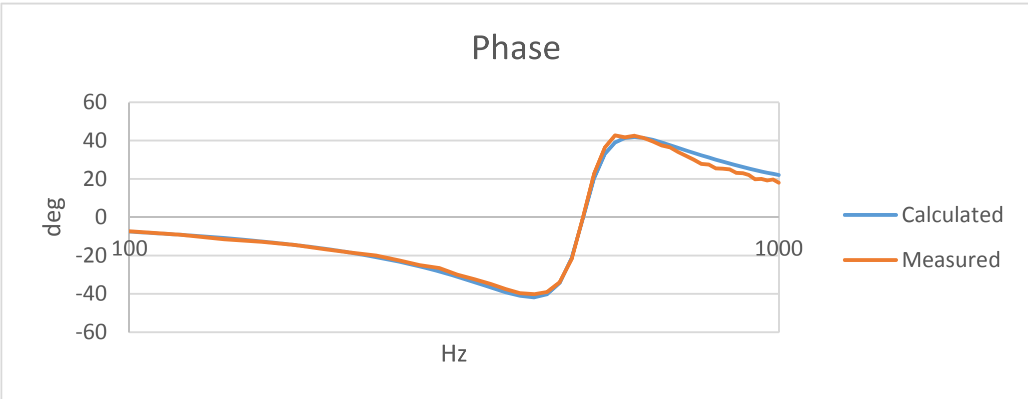
$$\omega_{-3dB}=\frac{2}{T}\,tan(\omega_{-3dB}{}^*\,T/2)=\frac{2}{250\times10^{-6}}\,tan(2\pi\times300Hz\ \times250\times10^{-6}/2)=1921rad/s$$

$$\omega_{bw}=(\omega_o-\omega_{-3dB})\times2=(3314-1921)\times2=2786rad/s$$

Bi-quad Filter1: a0 (04.051) = 0.8167
Bi-quad Filter1: a1 (04.052) = -1.0901
Bi-quad Filter1: a2 (04.053) = 0.7250
Bi-quad Filter1: b1 (04.054) = -1.0901
Bi-quad Filter1: b2 (04.055) = 0.5417

The results given below show the measured results compared with the calculated characteristic for an ideal notch filter with a centre frequency of 500Hz, a bandwidth of 400Hz and a gain of 0.2.





Lag-lead filter design

A lag-lead filter in the s domain is given by

$$H(s) = \frac{1 + s/(2\pi f_2)}{1 + s/(2\pi f_1)} = \frac{1 + s/\omega_2}{1 + s/\omega_1} = \frac{\omega_1}{\omega_2}\left(\frac{\omega_2 + s}{\omega_1 + s}\right)$$

The required lead and lag frequencies (ω_2* and ω_1* respectively) must first be modified to prevent frequency distortion.

$$\omega_1 = \frac{2}{T}\, tan(\omega_1{}^* T/2)$$

$$\omega_2 = \frac{2}{T}\, tan(\omega_2{}^* T/2)$$

Using the bilinear transform

$$H(z) = \frac{\omega_1\omega_2 + \omega_1\left(\frac{2}{T}\left(\frac{z-1}{z+1}\right)\right)}{\omega_1\omega_2 + \omega_2\left(\frac{2}{T}\left(\frac{z-1}{z+1}\right)\right)}$$

$$H(z) = \frac{(\omega_1\omega_2 T/2)(z+1) + \omega_1(z-1)}{(\omega_1\omega_2 T/2)(z+1) + \omega_2(z-1)}$$

$$H(z) = \frac{(\omega_1\omega_2 T/2 + \omega_1)z + \ (\omega_1\omega_2 T/2 - \omega_1)}{(\omega_1\omega_2 T/2 + \omega_2)z + \ (\omega_1\omega_2 T/2 - \omega_2)}$$

$$H(z) = \frac{\left(\frac{\omega_1\omega_2 T/2 + \omega_1}{\omega_1\omega_2 T/2 + \omega_2}\right)z + \left(\frac{\omega_1\omega_2 T/2 - \omega_1}{\omega_1\omega_2 T/2 + \omega_2}\right)}{z + \left(\frac{\omega_1\omega_2 T/2 - \omega_2}{\omega_1\omega_2 T/2 + \omega_2}\right)}$$

$$H(z) = \frac{a_0 z + a_1}{z + b_1}$$

And so

$$a_0 = \frac{\omega_1\omega_2 T/2 + \omega_1}{\omega_1\omega_2 T/2 + \omega_2}$$

$$a_1 = \frac{\omega_1\omega_2 T/2 - \omega_1}{\omega_1\omega_2 T/2 + \omega_2}$$

$$b_1 = \frac{\omega_1\omega_2 T/2 - \omega_2}{\omega_1\omega_2 T/2 + \omega_2}$$

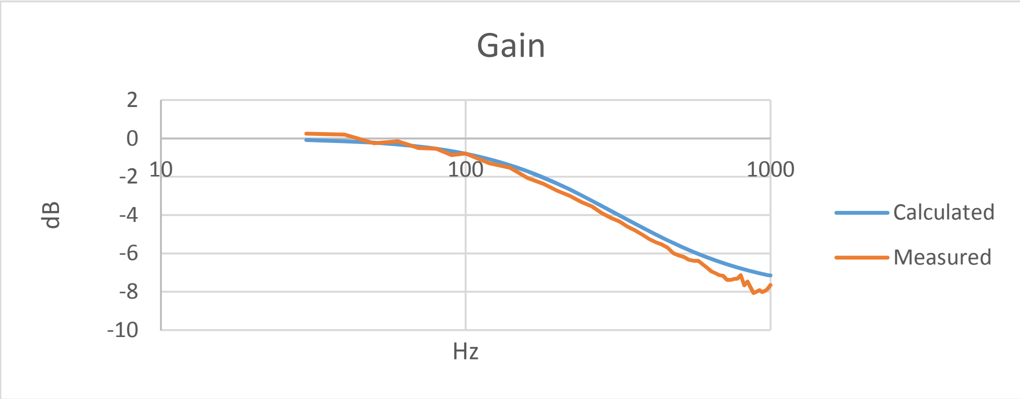
If the required filter characteristics are f_1=200Hz, f_2=500Hz then these frequencies must be modified to correct for frequency distortion.

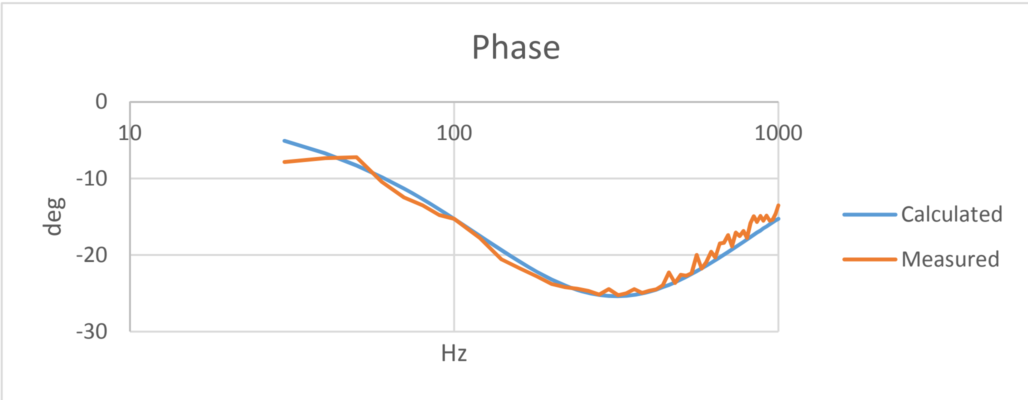
$$\omega_1 = \frac{2}{T}\, tan(\omega_1{}^{*}T/2) = \frac{2}{250 \times 10^{-6}}\, tan(2\pi \times 200Hz\ \times 250 \times 10^{-6}/2) = 1267 rad/s$$

$$\omega_2 = \frac{2}{T}\, tan(\omega_2{}^{*}T/2) = \frac{2}{250 \times 10^{-6}}\, tan(2\pi \times 500Hz\ \times 250 \times 10^{-6}/2) = 3314 rad/s$$

Bi-quad Filter1: a0 (04.051) = 0.4668
Bi-quad Filter1: a1 (04.052) = -0.1934
Bi-quad Filter1: a2 (04.053) = 0.0000
Bi-quad Filter1: b1 (04.054) = -0.7265
Bi-quad Filter1: b2 (04.055) = 0.0000

The results given below show the measured results compared with the calculated characteristic for an ideal lag-lead filter with f_1=200Hz and f_2=500Hz.





2nd order low pass Butterworth Filter

A 2nd order low pass Butterworth filter in the s domain is given by

$$H(s) = \frac{{\omega_{co}}^2}{s^2 + \sqrt{2}\omega_{co}s + {\omega_0}^2}$$

ω_{co} is the cut off frequency in rad/s.

The required cut off frequency $\omega_{co}{}^*$ must first be modified to prevent frequency distortion.

$$\omega_{co} = \frac{2}{T}\, tan({\omega_{co}}^*\, T/2)$$

Using the bilinear transform

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$$

$$H(z) = \frac{{\omega_{co}}^2}{\left(\frac{2}{T}\left(\frac{z-1}{z+1}\right)\right)^2 + \sqrt{2}\omega_{co}\left(\frac{2}{T}\left(\frac{z-1}{z+1}\right)\right) + {\omega_{co}}^2}$$

$$H(z) = \frac{\left(\frac{\omega_{co}T}{2}\right)^2 (z+1)^2}{(z-1)^2 + \sqrt{2}\left(\frac{\omega_{co}T}{2}\right)(z-1)(z+1) + \left(\frac{\omega_{co}T}{2}\right)^2 (z+1)^2}$$

$$H(z) = \frac{\left(\frac{\omega_{co}T}{2}\right)^2 z^2 + 2\left(\frac{\omega_{co}T}{2}\right)^2 z + \left(\frac{\omega_{co}T}{2}\right)^2}{z^2 - 2z + 1 + \sqrt{2}\left(\frac{\omega_{co}T}{2}\right)z^2 - \sqrt{2}\left(\frac{\omega_{co}T}{2}\right) + \left(\frac{\omega_{co}T}{2}\right)^2 z^2 + 2\left(\frac{\omega_{co}T}{2}\right)^2 z + \left(\frac{\omega_{co}T}{2}\right)^2}$$

$$H(z) = \frac{z^2\left(\frac{\omega_{co}T}{2}\right)^2 + z\left(2\left(\frac{\omega_{co}T}{2}\right)^2\right) + \left(\frac{\omega_{co}T}{2}\right)^2}{z^2\left(1+\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2\right) + z\left(2\left(\frac{\omega_{co}T}{2}\right)^2 - 2\right) + \left(1-\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2\right)}$$

$$H(z) = \frac{z^2\left(\frac{\left(\frac{\omega_{co}T}{2}\right)^2}{1+\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}\right) + z\left(\frac{2\left(\frac{\omega_{co}T}{2}\right)^2}{1+\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}\right) + \left(\frac{\left(\frac{\omega_{co}T}{2}\right)^2}{1+\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}\right)}{z^2 + z\left(\frac{2\left(\left(\frac{\omega_{co}T}{2}\right)^2 - 1\right)}{1+\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}\right) + \left(\frac{1-\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}{1+\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}\right)}$$

$$H(z) = \frac{a_0 z^2 + a_1 z + a_2}{z^2 + b_1 z + b_2}$$

And so

$$a_0 = \frac{\left(\frac{\omega_{co}T}{2}\right)^2}{1+\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}$$

$$a_1 = 2a_0$$

$$a_2 = a_0$$

$$b_1 = \frac{2\left(\left(\frac{\omega_{co}T}{2}\right)^2 - 1\right)}{1+\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}$$

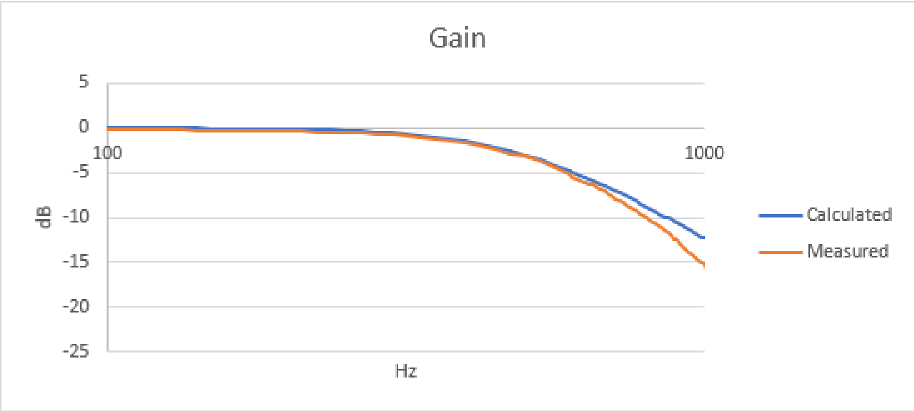
$$b_2 = \frac{1-\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}{1+\sqrt{2}\left(\frac{\omega_{co}T}{2}\right)+\left(\frac{\omega_{co}T}{2}\right)^2}$$

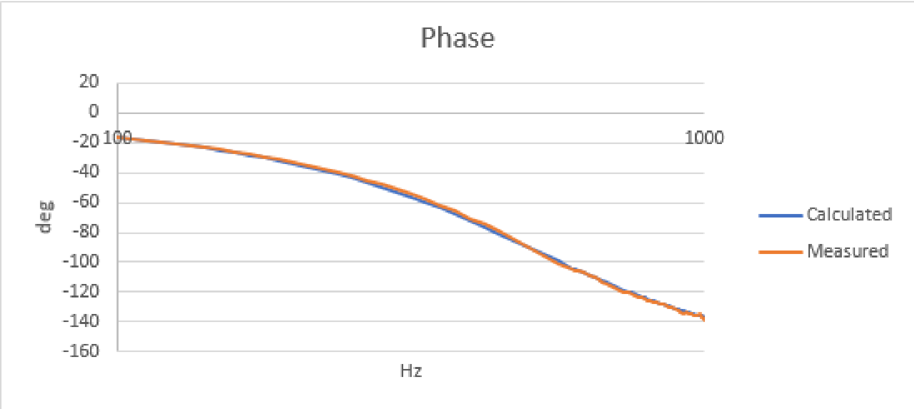
If the required cut-off frequency is 500Hz then ω_{co} is given by

$$\omega_{co} = \frac{2}{250 \times 10^{-6}}\, tan(2\pi \times 500Hz\ \times 250 \times 10^{-6}/2) = 3314 rad/s$$

Bi-quad Filter1: a0 (04.051) = 0.0976
Bi-quad Filter1: a1 (04.052) = 0.1953
Bi-quad Filter1: a2 (04.053) = 0.0976
Bi-quad Filter1: b1 (04.054) = -0.9428
Bi-quad Filter1: b2 (04.055) = 0.3333

The results given below show the measured results compared with the calculated characteristic for an ideal 2nd order low pass Butterworth filter with f_{co}=500Hz.


Gain
5
0
-5
-10
-15
-20
-25
dB
100
1000
Hz
Calculated
Measured


Phase
20
0
-20
-40
-60
-80
-100
-120
-140
-160
deg
100
1000
Hz
Calculated
Measured

Matlab test script

The following is a Matlab script that can be copied and pasted into the Matlab editor and then run to test digital filter designs. The version given here includes a notch filter and gives the continuous transfer function and the digital filter transfer function. The digital filter implementation is based on floating point values, but the implementation in the drive is based on integer maths. The results from a floating point or integer implementation are very similar, so this script gives a good approximation to the actual results.

```
% ***************************************************************************************
% Digital Filter Design Tester
%
% This module is used to test a digital filter design based on the bi-quad format with
% 3 x A and 2 x B coefficients. Results are produced for the original continuous
% transfer function and a digital filter implementation.
% **************************************************************************************
function DigitalFilterDesign()

    %*********************************************************************************
    % Set-up parameters
    %*********************************************************************************

    % Sample time.
    T = 250e-6;

    % Frequency range for testing the filter.
    FMin = 1;
    FMax = 2000;

    % Define the frequency resolution and calculate the number of frequency samples.
    FDelta = 5;
    N = round((FMax - FMin) / FDelta);

    %*********************************************************************************
    % Filter coefficients
    %*********************************************************************************

    % ***** Start of modification for different filter type *****

    % Define the filter characteristics. In this case it is a notch filter. f0=centre
    % frequency(Hz), fbw=bandwidth(Hz)and NotchGain=gain at the centre frequency.
    f0 = 500;
    fbw = 400;
    NotchGain = 0.2;

    % Centre frequency and bandwidth in rad/s.
    w0 = 2 * pi * f0;
    wbw = 2 * pi * fbw;

    % Compensate for frequency distortion.
    w0Mod = (2 / T) * tan(w0 * T / 2);
    w3dBLower = w0 - wbw / 2;
    w3dBLowerMod = (2 / T) * tan(w3dBLower * T / 2);
    wbwMod = 2 * (w0Mod - w3dBLowerMod);

    % Digital filter coefficients.
    w0_T_2Squared = (w0Mod * T / 2)^2;
    wbw_T_2 = wbwMod * T / 2;
    Denom = 1 + wbw_T_2 + w0_T_2Squared;

    a0 = (1 + NotchGain * wbw_T_2 + w0_T_2Squared) / Denom;
    a1 = -2 * (1 - w0_T_2Squared) / Denom;
    a2 = (1 - NotchGain * wbw_T_2 + w0_T_2Squared) / Denom;
    b1 = a1;
    b2 = (1 - wbw_T_2 + w0_T_2Squared) / Denom;

    % ***** End of modification for different filter type *****

    % Display the calculated coefficients.
```

```
fprintf('a0=%.4f\n', a0);
fprintf('a1=%.4f\n', a1);
fprintf('a2=%.4f\n', a2);
fprintf('b1=%.4f\n', b1);
fprintf('b2=%.4f\n', b2);

%*********************************************************************************
% Main loop
% The gain and phase are calculated for the continuous transfer function with the
% closed-form solution. For the digital implementation the filter is tested by
% injecting 1000 cycles of the test frequency and then correlating the output
% with the injection frequency. This gives the gain and phase in the frequency
% domain.
%*********************************************************************************

% Results arrays for gain/phase characteristics of the continuous and digital
% filter.
Frequency = zeros(N);
GainContinuous = zeros(N);
PhaseContinuous = zeros(N);
GainDigital = zeros(N);
PhaseDigital = zeros(N);

% Loop once for each test frequency.
for n = 1 : N

    % Test frequency
    f = FMin + (n - 1) * FDelta;
    w = 2 * pi * f;
    Frequency(n) = f;

    % =========================================
    % Continuous filter transfer function
    % =========================================

    % ***** Start of modification for different filter type *****

    Numerator = complex(1 - (w / w0)^2, NotchGain * (wbw / w0^2) * w );
    Denominator = complex(1 - (w / w0)^2, (wbw / w0^2) * w);
    G = Numerator / Denominator;

    % ***** End of modification for different filter type *****

    % Convert to gain (dB) and phase (deg).
    GainContinuous(n) = 20 * log10(abs(G));
    PhaseContinuous(n) = angle(G) * 180 / pi;

    % =========================================
    % Sampled filter transfer functions
    % =========================================

    % Floating point state values.
    In_1 = 0;
    In_2 = 0;
    On_1 = 0;
    On_2 = 0;

    % Floating point correlation accumulators.
    SinAcc = 0;
    CosAcc = 0;

    % Test with 1000 cycles of the test frequency.
    % Number of samples = 1000 x TTest / TSample = 100 x (1 / f) / T.
    NumberOfSamples = 1000 / (f * T);
    for m = 1 : NumberOfSamples

        % Test signal.
        In = sin(w * T * m);
```

```
            % Floating point implementation.
            % ==============================
            On = a0 * In + a1 * In_1 + a2 * In_2 - b1 * On_1 - b2 * On_2;
            In_2 = In_1;
            In_1 = In;
            On_2 = On_1;
            On_1 = On;

            % Accumulate the correlated sine and cosine components.
            SinAcc = SinAcc + On * sin(w * T * m);
            CosAcc = CosAcc + On * cos(w * T * m);
        end

        % Rescale the components to have the same scaling as the unity magnitude test
        % signal.
        SinAcc = 2 * SinAcc / NumberOfSamples;
        CosAcc = 2 * CosAcc / NumberOfSamples;

        GainDigital(n) = 20 * log10(sqrt(SinAcc^2 + CosAcc^2));
        PhaseDigital(n) = atan(CosAcc / SinAcc) * 180 / pi;
    end

    %*******************************************************************************
    % Plot results
    % A gain (dB) and phase (deg) plot are produced one above the other.
    %*******************************************************************************

    % Gain plot showing continuous and digital filter gain with log frequency axis.
    subplot(2,1,1)
    semilogx(Frequency, GainContinuous, 'r', Frequency, GainDigital, 'b');
    grid on;
    xlabel('Frequency(Hz)');
    ylabel('Gain');

    % Phase plot showing continuous and digital filter gain with log frequency axis.
    subplot(2,1,2)
    semilogx(Frequency, PhaseContinuous, 'r', Frequency, PhaseDigital, 'b');
    grid on;
    xlabel('Frequency(Hz)');
    ylabel('Phase(deg)');

    % Indicate which plot is continuous and which is digital.
    fprintf('Red = continuous, Blue = digital\n');
end
```