Designing a Digital Notch Filter

Digital Signal Processing

Notes

Contents

Notes

Notch Filter

Problem:

- "*Cleaning up*" an audio signal
- 50Hz mains hum (60Hz in USA)
- Digital filter with gain = 0 @ 50Hz
- gain = 1 for all other frequencies.

Known as a Notch Filter

Notes

Design Approach

Step 1:

- Design continuous time analogue filter that fits specification

Step 2:

- Use bilinear transformation to convert to digital filter

Result:

- Filter that uses past outputs as well as inputs when calculating current output.

Known as recursive filters, autoregressive filters or Infinite Impulse Response (IIR) filters.

Notes

50Hz Notch Filter

Specification:

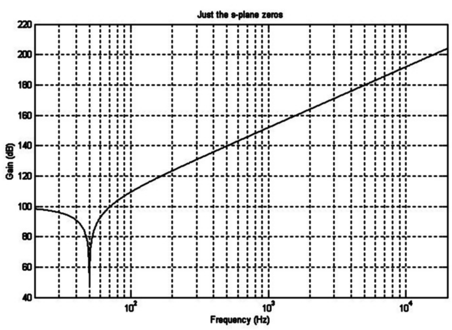
- Zero gain at some frequency f_0 (50 Hz in this case, so $\omega_0 = 100\pi$)
- Close to unity gain at $|f - f_0| >> 0$
- i.e. Overall amplitude response to be flat except close to f_0.

Notes

50Hz Notch Filter

1st requirement (0 gain @ 0Hz): s-domain zeros at $s = \pm j\omega_0$, which leads:

$$H_0(s) = (s - j\omega_0) \times (s + j\omega_0) = s^2 + \omega_0^2$$

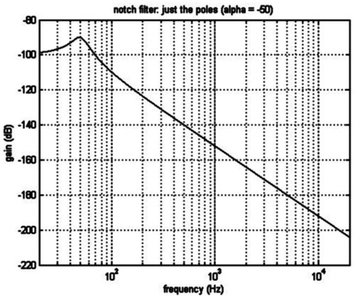


Notes

50Hz Notch Filter

2nd requirement, require poles as well as the zeros.
Placing at $s = \alpha \pm j\omega_0$, which introduces the following response:

$$H_p(s) = \frac{1}{(s-(\alpha+j\omega_0))(s-(\alpha-j\omega_0))} = \frac{1}{(s-\alpha)^2+\omega_0^2}$$

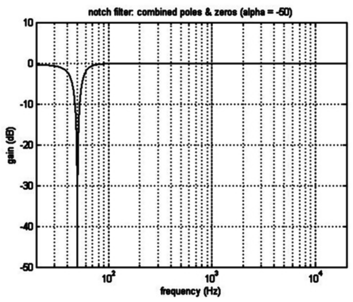


Notes

50Hz Notch Filter

At frequencies long way from notch $|s| >> |\omega_0|$, the poles & zeros effectively cancel each other out, as shown below:

$$H(s) = H_0(s) \times H_p(s) = \frac{s^2+\omega_0^2}{(s-\alpha)^2+\omega_0^2}$$



Similar to what we will transform into a digital filter.

Notes

50Hz Notch Filter

Given an analogue prototype

$$H_a(s) = \frac{s^2+\omega_0^2}{(s-\alpha)^2+\omega_0^2}$$

Design a digital notch filter using the bilinear transformation:

$$s \leftarrow \frac{2}{T}\frac{z-1}{z+1}$$

Giving:

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

Notes

50Hz Notch Filter

Multiplying through by $(z+1)^2$:

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

Going to result in a standard biquadratic filter.

Notes

50Hz Notch Filter

- Numerical values of the filter's coefficients depend on T, α and ω_0.
- For audio mains-hum removal filter, $f_s = 44100$ Hz, so $\frac{2}{T} = 88200$.
- To determine other values is more complicated.

Using bilinear transformation frequency warping formula:

$$\Omega_0 = 2\pi\frac{f}{f_s} = \frac{100\pi}{44100} = 7.1239... \times 10^{-3}$$

and thus

$$\omega_0 = \frac{2}{T} tan\left(\frac{\Omega_0}{2}\right) = 88200 \times 3.56191... \times 10^{-3}.$$

Notes

50Hz Notch Filter

Interesting warped notch frequency:

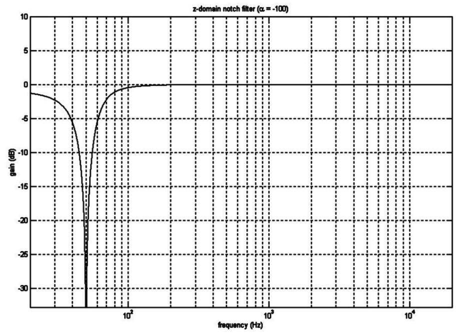
$$f_0 = \frac{\omega_0}{2\pi} = 50.00021145...Hz$$

- Almost identical to the notch frequency an analogue filter would need to do the same job,
- i.e. there is virtually no frequency warping (a difference of only 4 parts per million).
- This is because f_0 is very small compared to fs so the frequency warping effect is tiny.
- If we were looking for a notch up in the several-kHz region this similarity would not happen.

Notes

50Hz Notch Filter

MATLAB program notch_z.m provides plots to illustrate the effect of chosen values. e.g. value of $\alpha = -100$.



Notes

Summary

- Notch filter designed in the analog domain;
- Further design process to convert it to a digital form.

Notes

Notes