The bilinear z transform

Posted on March 2, 2003 by Nigel Redmon

The bilinear transform is the most popular method of converting analog filter prototypes in the s domain to the z domain so we can implement them as digital filters. The reason we are interested in these s domain filters is that analog filter theory has been around longer than digital filter theory, so we have available to us a number of popular and useful filters in the s domain. The standard biquads used heavily in audio processing are direct implementations from the s domain using the bilinear transform.

The theory of the bilinear transform is well documented in DSP texts and on the web, so rather than spend time going into the theory, we'll cut to the chase and show how to do the transform, using a second order lowpass filter as an example.

The s domain transfer function of a second order lowpass filter is

$$H(s) := \frac{1}{s^2 + \frac{s}{Q} + 1}$$

To move to the z domain, we need to substitute for s in terms of z. At the same time, we "pre-warp" the response using the tan function in order to wrap the infinite s plane onto the circular z plane. The combined substitution is

$$s := \frac{1}{K}\cdot\frac{z-1}{z+1} \qquad \text{where} \qquad K := \tan\left(\frac{\omega T}{2}\right) \qquad \text{and} \qquad \omega T := 2\cdot\pi\cdot\frac{Fc}{Fs}$$

Here's our z domain transfer function after substituting for s:

$$H(z) := \frac{1}{\left(\frac{1}{K}\cdot\frac{z-1}{z+1}\right)^2 + \frac{\frac{1}{K}\cdot\frac{z-1}{z+1}}{Q} + 1}$$

In order to solve for our filter coefficients, we need to get the equation into the form of a second order polynomial in the numerator (the zeros) and another second order polynomial in the denominator (the poles). That is, we need it to look something like this:

$$H(z) := \frac{a_0 + a_1\cdot z^{-1} + a_2\cdot z^{-2}}{b_0 + b_1\cdot z^{-1} + b_2\cdot z^{-2}}$$

It helps to use a math package to cut down on some of the tedious work. If we take the right side of the equation and ask MathCad to "simplify", it gives

$$K^2\cdot(z+1)^2\cdot\frac{Q}{\left(Q\cdot z^2 - 2\cdot Q\cdot z + Q + K\cdot z^2 - K + K^2\cdot Q\cdot z^2 + 2\cdot K^2\cdot Q\cdot z + K^2\cdot Q\right)}$$

The first thing we see is that there are too many Q's, so divide the top and bottom by Q, resulting in

$$\frac{K^2\cdot(z+1)^2}{z^2 - 2\cdot z + 1 + \frac{K}{Q}\cdot z^2 - \frac{K}{Q} + K^2\cdot z^2 + 2\cdot K^2\cdot z + K^2}$$

Now combine terms in the numerator and rearrange in descending powers of z; do the same for the denominator:

$$\frac{K^2\cdot z^2 + 2\cdot K^2\cdot z + K^2}{\left(K^2+\frac{K}{Q}+1\right)\cdot z^2 + 2\cdot\left(K^2-1\right)\cdot z + \left(K^2-\frac{K}{Q}+1\right)}$$

The z operator means delay, and its power tell us how many samples. Positive powers of z are in the future, and negative powers are in the past. Our filter can't see into the future (we call such filters "non-causal"), so we need to reduce the powers of z to zero and below (to make it causal). To do this we multiply the numerator and denominator by z^{-2}, which subtracts two from each power of z, and gives us our final form:

$$H(z) := \frac{K^2 + 2\cdot K^2\cdot z^{-1} + K^2\cdot z^{-2}}{\left(K^2+\frac{K}{Q}+1\right) + 2\cdot\left(K^2-1\right)\cdot z^{-1} + \left(K^2-\frac{K}{Q}+1\right)\cdot z^{-2}}$$

Our coefficients are now evident:

$$a_0 := K^2 \qquad a_1 := 2\cdot K^2 \qquad a_2 := K^2$$
$$b_0 := K^2+\frac{K}{Q}+1 \qquad b_1 := 2\cdot\left(K^2-1\right) \qquad b_2 := K^2-\frac{K}{Q}+1$$

Because the b0 coefficient is associated with y[n], the filter output, we divide everything by b0 to normalize the filter. Here we also note some convenient relationships between the "a" coefficients:

$$a_0 := \frac{K^2}{K^2+\frac{K}{Q}+1} \qquad a_1 := 2\cdot a_0 \qquad a_2 := a_0$$
$$b_0 := 1 \qquad b_1 := \frac{2\cdot\left(K^2-1\right)}{K^2+\frac{K}{Q}+1} \qquad b_2 := \frac{K^2-\frac{K}{Q}+1}{K^2+\frac{K}{Q}+1}$$

As a further efficiency, you can compute the common numerator term once and use it where needed. When calculating the coefficients, Q is the filter Q value (for a Butterworth lowpass, where the passband is flat with no corner peaking, use 0.7071, which is the reciprocal of the square root of two). Recall that we defined K as

$$K := \tan\left(\pi\cdot\frac{Fc}{Fs}\right)$$

where Fc is the filter corner frequency you want, and Fs is the sample rate.

This entry was posted in Digital Audio, Filters, IIR Filters. Bookmark the permalink.

23 Responses to *The bilinear z transform*

Alex Irae *says:*
June 5, 2013 at 8:19 am

Flawless explanation Nigel, excellent tutorials!

Reply

Robin Green *says:*
September 6, 2013 at 4:53 pm

The clearest explanations of the process I have ever read! I now get that it's kinda mechanical in Mathematica or Maple, yet requires human assistance to get right.

Any advice on approaching polynomials where y[0] isn't present? Do you timeshift the whole expression one sample and go from there or is there a different "correct thing" to do?