EE 342

IIR Filter Design

One way to design IIR filters is by use of the bilinear transformation. To design a low-pass IIR filter, transform a low-pass continuous-time filter to a discrete-time filter using the transformation:

$$s = \frac{1 - z^{-1}}{1 + z^{-1}} = \frac{z - 1}{z + 1}$$

To design another type of filter (high-pass, band-pass, or band-stop) another step is needed. There are several ways to do this. One of the simplest is to find different transformations which will map a low-pass continuous-time filter in one of these other types of discrete-time filters. Here are a set of transformations which will do that.

In the equations which follow, ω refers to continuous-time frequencies, and Ω refers to discrete-time frequencies.

	LP	HP	BP	BS
s	$\frac{z-1}{z+1}$	$\frac{z+1}{z-1}$	$\frac{z^2-2cz+1}{z^2-1}$	$\frac{z^2-1}{z^2-2cz+1}$
z	$\frac{1+s}{1-s}$	$\frac{s+1}{s-1}$	$\frac{-c\pm\sqrt{c^2+s^2-1}}{s-1}$	$\frac{c\pm\sqrt{c^2+\left(\frac{1}{s}\right)^2-1}}{1-\frac{1}{s}}$
ω	$\tan(\frac{\Omega}{2})$	$-\cot(\frac{\Omega}{2})$	$\frac{c-\cos(\Omega)}{\sin(\Omega)}$	$\frac{\sin(\Omega)}{c-\cos(\Omega)}$
c			$\frac{\sin(\Omega_{pl}+\Omega_{ph})}{\sin(\Omega_{pl})+\sin(\Omega_{ph})}$	$\frac{\sin(\Omega_{pl}+\Omega_{ph})}{\sin(\Omega_{pl})+\sin(\Omega_{ph})}$
Ω_1	0	π	$\cos^{-1}(c)$	0 or π

A filter is usually specified by passband and stopband frequencies Ω_{pass} and Ω_{stop}, and by passband ripple R_{pass}, and stopband attenuation R_{stop}. Band-pass and band-stop filters have low and high passband and stopband frequencies.

To design a discrete-time IIR filter, do the following:

1. Pre-warp the discrete-time frequencies to continuous-time frequencies using the row labeled ω:
   (a) If low-pass or high-pass, transform Ω_{pass} and Ω_{stop} to ω_{pass} and ω_{stop}.

   (b) If band-pass or band-stop, find c. Then transform Ω_{pl}, Ω_{ph}, Ω_{sl} and Ω_{sh} to ω_{pl}, ω_{ph}, ω_{sl} and ω_{sh}. Let $\omega_{pass} = |\omega_{pl}|$, and let $\omega_{stop} = \min(|\omega_{sl}|, |\omega_{sh}|)$

2. Design a continuous time low-pass filter using ω_{pass} and ω_{stop} from step 1.

3. Transform the continuous-time filter to a discrete-time filter using one of the following two methods:

   (a) Replace every s in the continuous-time transfer function as specified with the row labeled s. Do lots of algebra to get it into a usable form.

   (b) Map s-plane poles and zeros to corresponding z-plane poles and zeros using the equations in the row labeled z. (For band-pass and band-stop filters, every s-plane pole maps to two z-plane poles, and every s-plane zero maps to two z-plane zeros.) Don't forget s-plane zeros at infinity. For low-pass filters, an s-plane zero at infinity maps into a z-plane zero at -1. For high-pass filters, an s-plane zero at infinity maps into a z-plane zero at +1. For band-pass filters, an s-plane zero at infinity maps into a z-plane zero at +1 and a z-plane zero at -1. For band-stop filters, and s-plane zero at infinity maps into z-plane zeros at $e^{\pm j\Omega_c}$ where $\Omega_c = \cos^{-1}(c)$.

   Then find k_d by

$$k_d = \left| \frac{\prod_{k=1}^{N} \left(e^{j\Omega_1} - p_{dk}\right)}{\prod_{k=1}^{N} \left(e^{j\Omega_1} - z_{dk}\right)} \right|$$

   where Ω_1 is the frequency where $|H(\Omega_1)| = 1$.

   The discrete-time transfer function is:

$$H(z) = k_d \frac{\prod_{k=1}^{N} (z - z_{dk})}{\prod_{k=1}^{N} (z - p_{dk})}$$

Using MATLAB, steps 1 and 2 can be done with the function `buttord`, `cheby1ord`, `cheby2ord` or `ellipord`. The entire design process can be completed with the function `butter`, `cheby1`, `cheby2` or `ellip`.