# Frequency Transformations For Analog Filters

| Type of Transformation | Transformation | Cutoff Frequencies of New Filter |
| --- | --- | --- |
| Lowpass | $s \to \frac{\Omega_p}{\Omega'_p} s$ | $\Omega'_p$ |
| Highpass | $s \to \frac{\Omega_p \Omega'_p}{s}$ | $\Omega'_p$ |
| Bandpass | $s \to \Omega_p \left( \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)} \right)$ | $\Omega_l, \Omega_u$ |
| Bandstop | $s \to \Omega_p \left( \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_l \Omega_u} \right)$ | $\Omega_l, \Omega_u$ |

Strategy:

(1) Prewarp the cutoff frequencies
(2) Design an analog lowpass filter
(3) Use a frequency transformation to prewarped frequencies
(4) Use the bilinear transform to get a digital filter

## Frequency Transformations For Digital Filters

| Type of Transformation | Transformation | Cutoff Frequencies of New Filter |
|---|---|---|
| Lowpass | $z^{-1} \rightarrow \frac{z^{-1} - a}{1 - az^{-1}}$ | $\omega'_p$<br>$a = \frac{\sin[(\omega_p - \omega'_p)/2]}{\sin[(\omega_p + \omega'_p)/2]}$ |
| Highpass | $z^{-1} \rightarrow -\frac{z^{-1} + a}{1 + az^{-1}}$ | $\omega'_p$<br>$a = -\frac{\cos[(\omega_p + \omega'_p)/2]}{\cos[(\omega_p - \omega'_p)/2]}$ |
| Bandpass | $z^{-1} \rightarrow -\frac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}$ | $\omega_l, \omega_u$<br>$a_1 = -2\alpha K/(K+1)$<br>$a_2 = (K-1)/(K+1)$<br>$\alpha = -\frac{\cos[(\omega_u + \omega_l)/2]}{\cos[(\omega_u - \omega_l)/2]}$<br>$K = \cot\left(\frac{\omega_u - \omega_l}{2}\right)\tan\left(\frac{\omega_p}{2}\right)$ |
| Bandstop | $z^{-1} \rightarrow \frac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}$ | $\omega_l, \omega_u$<br>$a_1 = -2\alpha/(K+1)$<br>$a_2 = -(K-1)/(K+1)$<br>$\alpha = \frac{\cos[(\omega_u + \omega_l)/2]}{\cos[(\omega_u - \omega_l)/2]}$<br>$K = \tan\left(\frac{\omega_u - \omega_l}{2}\right)\tan\left(\frac{\omega_p}{2}\right)$ |

Strategy:

(1) Design a digital lowpass filter (using standard techniques)

(2) Use a digital frequency transformation

## Designs of IIR Filters Based on Least Squares Methods

### (1) Padé's Approximation

Suppose $h_d(n)$ is specified for $n \geq 0$. Our desired filter has the form:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} = \sum_{k=0}^{\infty} h(k) z^{-k}$$

To find the best $\{a_k\}$ and $\{b_k\}$, we can minimize the squared error:

$$E = \sum_{n=0}^{U} [h_d(n) - h(n)]^2$$

Consider the case where $U = L - 1 = M + N - 1$:

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) - \ldots + b_0 x(n) + b_1 x(n-1) + \ldots + b_M x(n-M)$$

This implies:

$$h(n) = -a_1 h(n-1) - a_2 h(n-2) - \ldots - a_N h(n-N) + b_0 \delta(n) + b_1 \delta(n-1) + \ldots + b_M \delta(n-M)$$

Since $\delta(n-k) = 0$ except for $n = k$,

$$h(n) = -a_1 h(n-1) - a_2 h(n-2) - \ldots - a_N h(n-N) + b_n \qquad 0 \leq n \leq M$$

$$h(n) = -a_1 h(n-1) - a_2 h(n-2) - \ldots - a_N h(n-N) \qquad n > M$$

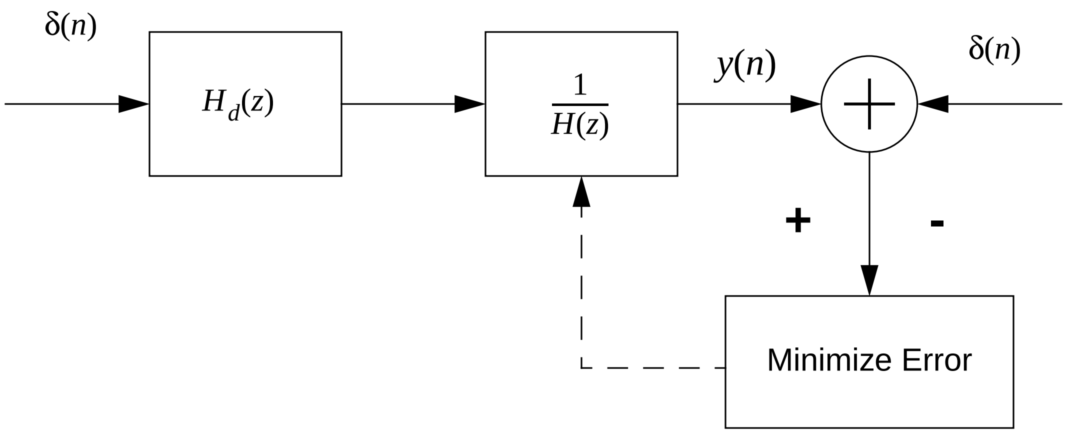
This gives a set of $N + M + 1$ linear equations. We can solve these equations using a standard linear equation solver (or matrix algebra) — this is called the *Padé approximation method*.

## (2) Least-Squares Using an All-Pole Filter

Suppose $h_d(n)$ is specified for $n \geq 0$. Our desired filter has the form:

$$H(z) = \frac{b_0}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

Consider the following minimization problem:




$$y(n) = \frac{1}{b_0}\left[h_d(n) + \sum_{k=1}^{N} h_d(n-k)\right]$$

Define the mean-squared error as:

$$E = \sum_{n=1}^{\infty} y^2(n)$$

where $y_d(0) = y(0) = 1$ implies $b_0 = h_d(0)$.

Differentiating w.r.t. $a_k$ and rewriting as a system of linear equations:

$$\sum_{k=1}^{N} a_k r_{dd}(k, l) = -r_{dd}(l, 0) \qquad l = 1, 2, \ldots, N$$

where, for finite data $(L \gg N)$,

$$r_{dd}(k, l) = r_{dd}(k-l) = \sum_{n=0}^{L-|k-l|} h_d(n) h_d(n+k-l) \qquad 0 \leq k-l \leq N$$

## (3) Least-Squares Using an ARMA (Pole/Zero) Filter

Suppose our desired filter has the form:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

The impulse response can be written as:

$$h(n) = -\sum_{k=1}^{N} a_k h(n-k) + \sum_{k=0}^{M} b_k \delta(n-k) \qquad n \geq 0$$

or,

$$h(n) = -\sum_{k=1}^{N} a_k h(n-k) + b_n \qquad 0 \leq n \leq M$$

$$= -\sum_{k=1}^{N} a_k h(n-k) \qquad n > M$$

This is, in general, a nonlinear optimization problem (that can be solved directly with iterative algorithms).

A sub-optimal approach, known as Prony's method, uses a two-step process: (1) fit the poles of the filter; (2) minimize the remaining error using zeros.

Define:

$$\hat{h}_d(n) = -\sum_{k=1}^{N} a_k h_d(n-k) \qquad n > M$$

and,

$$E_1 = \sum_{n=M+1}^{\infty} [h_d(n) - \hat{h}_d(n)]^2$$

This gives:

$$\sum_{l=1}^{N} a_l r_{dd}(k, l) = -r_{dd}(k, 0)$$

where,

$$r_{dd}(k, l) = \sum_{n=M+1}^{\infty} h_d(n-k) h_d(n-l)$$

Once we solve this system of equations for $\{a_k\}$, we can find $\{b_k\}$ by solving:

$$b_n = h_d(n) + \sum_{k=1}^{N} a_k h_d(n-k) \qquad 0 \le n \le M$$

We can use the Padé approximation method to solve this problem.

## (4) Least-Squares Using a Two-Stage Optimization

In a two-step process, we optimize both the poles and zeros of the filter.

Consider the all-pole portion of the filter:

$$H_1(z) = \frac{1}{1 + \sum_{k=1}^{N} \hat{a}_k z^{-k}}$$

The difference equation corresponding this system is:

$$v(n) = -\sum_{k=1}^{N} \hat{a}_k v(n-k) + \delta(n) \qquad n \ge 0$$

$$v(n) = -\sum_{k=1}^{N} \hat{a}_k v(n-k) + \delta(n) \qquad n \geq 0$$

Similarly, the all-zero portion of the filter can be written as:

$$H_2(z) = \sum_{k=0}^{M} b_k z^{-k}$$

The impulse response can be written as:

$$\hat{h}_d(n) = \sum_{k=0}^{M} b_k v(n-k)$$

The error signal is given by:

$$e(n) = h_d(n) - \hat{h}_d(n) = h_d(n) - \sum_{k=0}^{M} b_k v(n-k)$$

Its energy is given by:

$$E_2 = \sum_{n=0}^{\infty} \left[ h_d(n) - \sum_{k=0}^{M} b_k v(n-k) \right]^2$$

By differentiating w.r.t. the filter coefficients, we obtain:

$$\sum_{k=0}^{M} b_k r_{vv}(k, l) = r_{hv}(l)$$

where,

$$r_{vv}(k, l) = \sum_{n=0}^{\infty} v(n-k)v(n-l) \quad \text{and} \quad r_{hv}(k) = \sum_{n=0}^{\infty} h_d(n)v(n-k)$$

The above equation can be solved using any standard technique (matrix inversion, Gaussian elimination, etc.).

This method of two-stage optimization is known as *Shanks' method*.