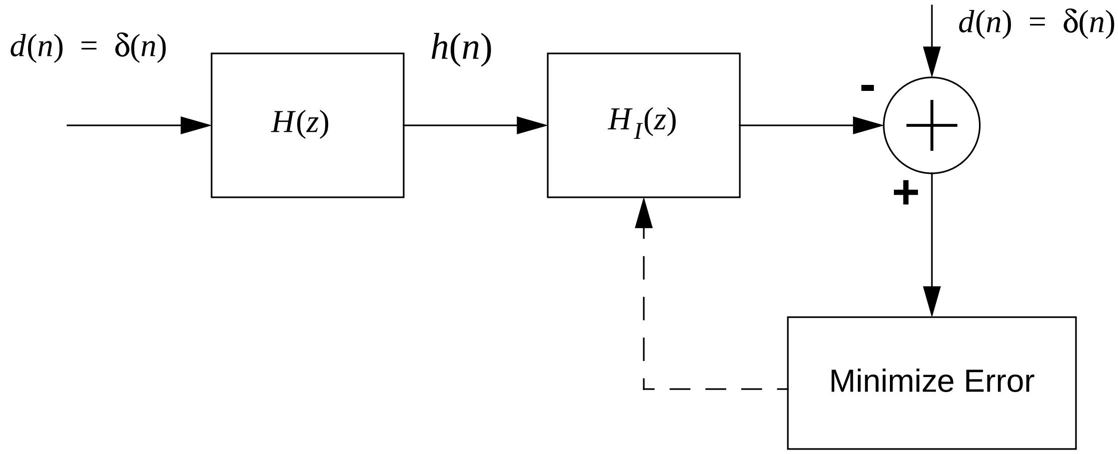
FIR Least-Squares Inverse Filters (Weiner Filters)

Consider the following optimization problem:



We seek to design an FIR filter, $H_I(z)$, such that:

$$h(n) \otimes h_I(n) = \delta(n)$$
$$H(z)H_I(z) = 1$$

$h_I(n)$ is an FIR filter: $H_I(z) = b_0 + b_1 z^{-1} + \ldots + b_M z^{-M} = \sum_{k=0}^{M} b_k z^{-k}$

We will use a least-squares error criterion:

$$e(n) = d(n) - \sum_{k=0}^{M} b_k z^{-k}$$
$$E = \sum_{n=0}^{\infty} e^2(n) = \sum_{n=0}^{\infty} \left(d(n) - \sum_{k=0}^{M} b_k z^{-k} \right)^2$$

Upon differentiating with respect to b_k:

$$\sum_{k=0}^{M} b_k r_{hh}(k-l) = r_{dh}(l) \qquad l = 0, 1, \ldots, M$$

where,

$$r_{hh}(l) = \sum_{n=0}^{\infty} h(n)h(n-l) \qquad \text{(autocorrelation)}$$

$$r_{dh}(l) = \sum_{n=0}^{\infty} d(n)h(n-l) \qquad \text{(crosscorrelation)}$$

The filter that satisfies this equation is called a *Weiner* filter.

If $d(n) = \delta(n)$,

$$r_{dh}(l) = \begin{cases} h(0) & l = 0 \\ 0 & \text{otherwise} \end{cases}$$

This gives the following system of equations:

$$\begin{bmatrix} r_{hh}(0) & r_{hh}(1) & \ldots & r_{hh}(M-1) & r_{hh}(M) \\ r_{hh}(1) & r_{hh}(0) & \ldots & r_{hh}(M-2) & r_{hh}(M-1) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ r_{hh}(M-1) & r_{hh}(M-2) & \ldots & r_{hh}(2) & r_{hh}(1) \\ r_{hh}(M) & r_{hh}(M-1) & \ldots & r_{hh}(1) & r_{hh}(0) \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \ldots \\ b_{M-1} \\ b_M \end{bmatrix} = \begin{bmatrix} h(0) \\ 0 \\ \ldots \\ 0 \\ 0 \end{bmatrix}$$

Solving this equation for $\{b_k\}$, and substituting the result into our expression for the mean-squared error gives:

$$E_{min} = 1 - h(0)b_0$$

Example 8.4.6: Find a 2nd order FIR approximation to:

$$h(n) = \begin{cases} 1 & n = 0 \\ -\alpha & n = 1 \\ 0 & \text{otherwise} \end{cases}$$

Method 1: Truncate the exact inverse:

$$H(z) = 1 - \alpha z^{-1}$$

$$H_I(z) = \frac{1}{H(z)} = \frac{1}{1 - \alpha z^{-1}} = 1 + \alpha z^{-1} + \alpha^2 z^{-2} + \ldots$$

Truncating $H_I(z)$ to two terms gives:

$$\hat{H}_I(z) = 1 + \alpha z^{-1}$$

The corresponding error is:

$$E_t = \sum_{n=2}^{\infty} \alpha^{2n} = \frac{\alpha^4}{1 - \alpha^2}$$

Method 2: Least squares solution:

$$r_{hh}(0) = 1 + \alpha^2 \qquad r_{hh}(1) = -\alpha \qquad h(0) = 1$$

therefore,

$$\begin{bmatrix} 1 + \alpha^2 & -\alpha \\ -\alpha & 1 + \alpha^2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and,

$$b_0 = \frac{1 + \alpha^2}{1 + \alpha^2 + \alpha^4} \qquad b_1 = \frac{\alpha}{1 + \alpha^2 + \alpha^4}$$

Note that:

$$E_{min} = 1 - h(0) b_0 = \frac{\alpha^4}{1 + \alpha^2 + \alpha^4} < E_t$$

Design of IIR Filters in the Frequency Domain

Consider the problem of fitting a general rational polynomial to a desired frequency response:

$$H(z) = G\prod_{k=1}^{K}\frac{1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2}}{1+\alpha_{1k}z^{-1}+\alpha_{2k}z^{-2}}$$

We must determine $\{a_k\}$, $\{b_k\}$, and G.

The frequency response can be expressed as:

$$H(\omega) = GA(\omega)e^{j\Theta(\omega)}$$

where

$$A(\omega) = \prod_{k=1}^{K}\frac{1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2}}{1+\alpha_{1k}z^{-1}+\alpha_{2k}z^{-2}}$$

and $\Theta(\omega)$ is the phase response.

For this type of problem, it is easier to deal with the group delay:

$$\tau_g(\omega) = -\frac{d}{d\omega}\Theta(\omega) = \tau_g(z)\Big|_{z=e^{j\omega}}\frac{dz}{d\omega}$$

It can be shown that:

$$\tau_g(z) = Re\left\{\prod_{k=1}^{K}\left[\frac{\beta_{1k}z+2\beta_{2k}}{z^2+\beta_{1k}z+\beta_{2k}}-\frac{\alpha_{1k}z+2\alpha_{2k}}{z^2+\alpha_{1k}z+\alpha_{2k}}\right]\right\}$$

At a selected group of frequencies, ω_k, the magnitude and phase errors are given by:

$$E_{mag} = GA(\omega_k) - A_d(\omega_k)$$

$$E_{phase} = \tau_g(\omega_k) - \tau_g(\omega_0) - \tau_d(\omega_k)$$

where $\tau_g(\omega_0)$ is the group delay in the passband of the filter (a minor detail).

If we define an objective function, or cost function as:

$$E(\bar{p}, G) = (1-\lambda)\sum_{n=1}^{L} w_n[GA(\omega_n) - A_d(\omega_n)]^2 + \lambda\sum_{n=1}^{L} v_n[\tau_g(\omega_n) - \tau_g(\omega_0) - \tau_d(\omega_n)]^2$$

we can optimize the filter design as per a user's requirements.

The optimal gain is given by:

$$\hat{G} = \frac{\sum_{n=1}^{L} w_n A(\omega_n) A_d(\omega_n)}{\sum_{n=1}^{L} w_n A^2(\omega_n)}$$

The error for this value of the gain is:

$$E(\bar{p}, G) = (1-\lambda)\sum_{n=1}^{L} w_n[\hat{G}A(\omega_n) - A_d(\omega_n)]^2 + \lambda\sum_{n=1}^{L} v_n[\tau_g(\omega_n) - \tau_g(\omega_0) - \tau_d(\omega_n)]^2$$

A closed-form solution for this equation is not possible since it involves a nonlinear optimization. A standard technique to solve this equation is to iterate on a solution starting with an initial guess. The iterative solution takes on the form:

$$\bar{p}^{(m+1)} = \bar{p}^{(m)} - \Delta^{(m)} Q^{(m)} g^{(m)}$$

This is a gradient search algorithm that iterates by adjusting the new values of the coefficients based on the amount of error and the derivatives in the neighborhood of the current solution.

Does it work?