ELECTRONOTES
1 Pheasant Lane
Ithaca, NY 14850
(607)-273-8030

APPLICATION NOTE NO. 318

February 1992

TIME DOMAIN LEAST SQUARED LOW-PASS FILTERS

In AN-315 we looked at a least squared error method of FIR filter design. For this method, we minimized the integrated squared error over regions in the frequency domain. Here we will be minimizing the squared error over a set of discrete points in the time domain. This is just the classical method of fitting a curve to a set of given points, for the case where the curve is of too low an order to fit all the data exactly. For example, we will want to find a reasonable fit of a straight line (first-order) to three points, while a second-order curve would be required for an exact fit. Accordingly, while involving a least squared error procedure, this method is likely most directly related to the exact curve fitting of AN-317. As in AN-317, the result will come out as a FIR filtering operation.

Fig. 1 shows our setup. We have three samples x(0), x(1), and x(2). We wish to fit a line to these points. This line has the equation $x'(t) = mt + b$. The parameters m and b will be set so that the total squared error, $E^2 = E_0^2 + E_1^2 + E_2^2$ is minimized. Once we find this line, we can use points on it too fill in missing samples, or to replace existing points if we wish. For example, the point x'(1) could be used to replace x(1). We will want to see what this does for us.

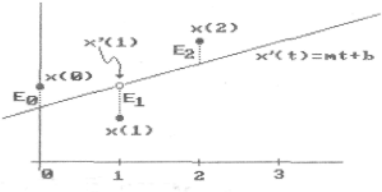

Fig. 1 Least Square Error Fit

Note that the errors E_0, E_1, and E_2 are given by:

$$E_0 = x(0) - b \tag{1a}$$

$$E_1 = x(1) - m - b \tag{1b}$$

$$E_2 = x(2) - 2m - b \tag{1c}$$

Accordingly, the squared error is:

$$E^2 = [\ x(0)^2 - 2bx(0) + b^2\ ]$$

$$+ [\ x(1)^2 - 2mx(1) - 2bx(1) + 2mb + m^2 + b^2\ ]$$

$$+ [\ x(2)^2 - 4mx(2) - 2bx(2) + 4m^2 + 4mb + b^2\ ] \quad (2)$$

We want to find values of m and of b that minimize E^2. Thus we need to take partial derivatives

$$\partial(E^2)/\partial m = -\ 2x(1) + 2b + 2m - 4x(2) + 8m + 4b$$

$$= -\ 2x(1) - 4x(2) + 6b + 10m = 0 \quad (3a)$$

$$\partial(E^2)/\partial b = -\ 2x(0) + 2b - 2x(1) + 2m + 2b - 2x(2) + 4m + 2b$$

$$= -\ 2x(0) - 2x(1) - 2x(2) + 6b + 6m = 0 \quad (3b)$$

Equations (3a) and (3b) can be solved to get:

$$m = [\ x(2) - x(0)\ ]\ /\ 2 \quad (4)$$

$$b = 5x(0)/6 + x(1)/3 - x(2)/6 \quad (5)$$

At this point, we have done nothing that is not readily available in many math text books. We have simply found a reasonable way to fit a line to three points. Next we need to bring in some signal processing - the idea of digital filtering. In particular, we will compute the value of x'(1), and suppose that we will be replacing x(1) with x'(1) as a means of digital filtering.

Since the straight line is:

$$x'(t) = mt + b \quad (6)$$

we have x'(1) = m + b, or:

$$x'(1) = [\ x(2)/2 - x(0)/2\ ] + [\ 5x(0)/6 + x(1)/3 - x(2)/6\ ]$$

$$= [\ x(0) + x(1) + x(2)\ ]\ /\ 3 \quad (7)$$

This is clearly a three-tap moving-average, a known low-pass filter. *In fact, it is known that the least square process gives a result at* the mean of t (which is t=1 for this case) that is the mean of x(t) (which is what equation (7) gives).

We can think of the least square fitting as a type of interpolation. As expected, we find a corresponding low-pass filtering. We could also solve for x'(t) at any other value of t, for example at t=1.5, at t=$\sqrt{2}$, or at t=7.

While we expect relatively little in the way of performance from this sort of design, it does illustrate the least-square error process that is so important in many formulations of digital signal processing.