FUNDAMENTALS OF PREDICTIVE FILTERING

RAFAEL B. CRUZ*

THE SEISMIC TRACE MODEL

When a dynamite explosion occurs at the bottom of a shot hole, this explosion creates white signal (i.e., signal containing all frequencies). However, since the earth is not a perfectly elastic medium, it distorts the signal, and selectively passes only a certain band of frequencies. This band-limited signal can be represented as an *average wavelet* (the sum of all the signal frequencies selectively passed by the Earth).

As this average wavelet travels into the Earth, part of the energy is reflected back towards the surface, at each layer interface, while the rest of the energy is transmitted deeper into the Earth.

We can represent this portion of reflected energy from each interface as a series of reflection coefficients at the vertical two-way travel time to each interface. Such a series of reflection coefficients we will term a *reflectivity sequence.*

If the "wavelet" travelling into the Earth were a sharp spike, the arrivals at the surface from the different reflecting horizons would look like our reflectivity sequence; i.e., a series of sharp spikes occurring at the two-way travel time to each interface.

However, since the signal going into the ground is in the form of an average wavelet, the reflected arrivals appear as a series of partially overlapping wavelets, each starting at the two way time for the particular reflecting interface.

This process of building the seismic trace as a series of partially overlapping wavelets can be considered as the "convolution" of the average wavelet going into the Earth with the reflectivity sequence representing the amount of energy reflected at each layer interface.

To further obscure the definition of each interface, we have noise added to the trace. We will classify this noise in three different categories: random noise, time-periodic noise, and space-coherent noise. This classification is made since different processing techniques will be employed to remove any of these types of noise. (Note that time-periodic noise can also be space-coherent). This seismic trace model is described in Figure 1.

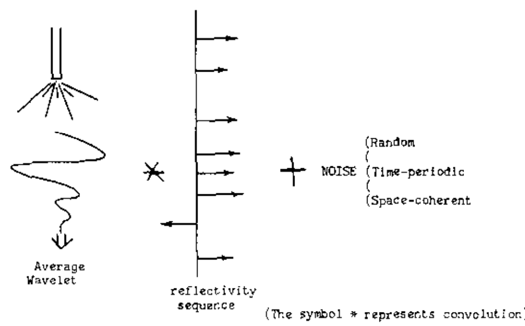


FIGURE 1. THE SEISMIC TRACE

The ultimate objective of seismic processing is to obtain, as close as possible, an estimate of what the reflectivity sequence looks like. Obviously, if we could represent each layer interface by a sharp spike, it would be possible to map these interfaces very accurately; thus obtaining maximum resolution.

By performing certain operations upon the recorded traces, we obtain a better approximation of the reflectivity sequence representing the Earth. In order to under-

*R. B. Cruz & Associates Ltd., Calgary, Alberta.

stand the processes that are applied to the seismic data, it is necessary to become familiar with some of the fundamental concepts of signal theory, the most important of which is *convolution.*

Let's consider any type of linear system, and input into that system a sharp unit spike or impulse. The output that we obtain from that linear system as a result of a sharp spike input is called the *impulse response* of that linear system. See Figure 2.

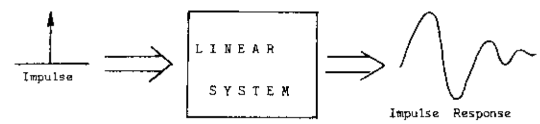


FIGURE 2. IMPULSE RESPONSE OF A LINEAR SYSTEM

Now let's consider an arbitrary input to the same linear system. The output of the system in this case is said to be the *convolution* of that arbitrary input with the

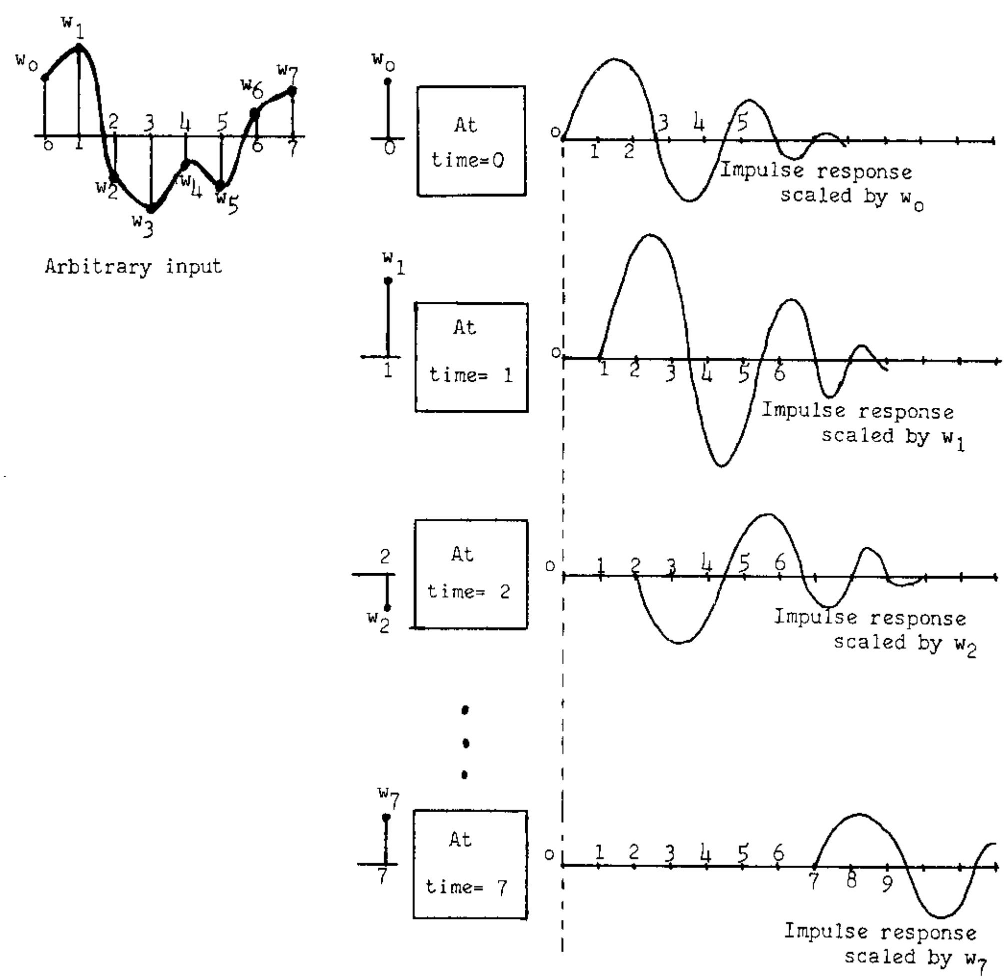


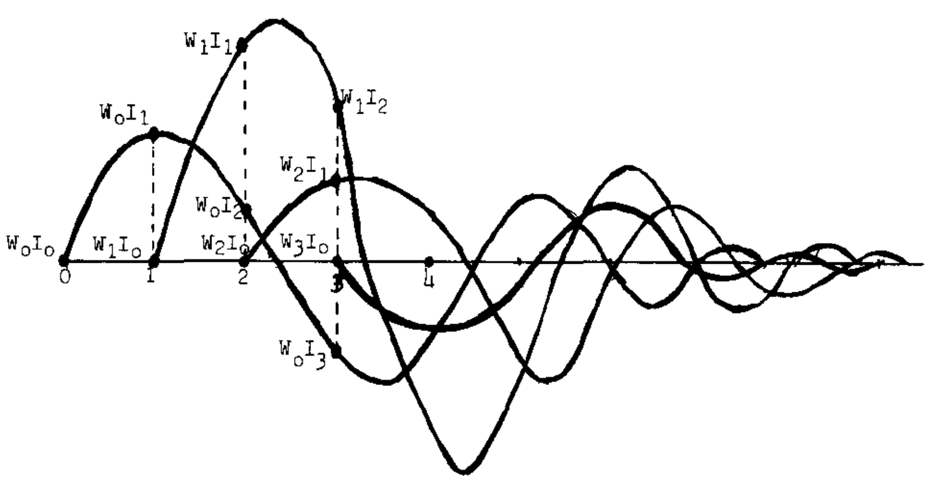
FIGURE 3. OUTPUT OF A LINEAR SYSTEM TO ARBITRARY INPUT SHOWING DELAYED IMPULSE RESPONSES GENERATED BY DIFFERENT TIME VALUES OF INPUT

impulse response of the system. Let's see what we mean by this.

Consider that arbitrary input to be a discretely sampled wavelet at equal intervals of time (for simplicity let's assume that interval to be one time unit). Now, we can consider the arbitrary input to be composed of a series of spikes, each representing the amplitude of the input wavelet at the particular sample time.

If we now view this operation as occurring in "real time", we have at time '0' the first sampled value of the input (W_0) entering the system and generating an output impulse response starting at time '0'. One unit of time later, i.e., at time '*1*', the next sampled value of the input (W_1) enters the system, thus generating a new impulse response starting at time '*1*'. Similarly, as new values of the input wavelet enter the system, they will generate impulse responses starting at later times. Obviously, the shape of the output will be described by the cumulative effect of all these impulse responses. The process is pictorially described in Figure 3.

If we look at the composite output more closely, we will be able to describe each final output value as the sum of different time values of the impulse response, each scaled by the input value that generated it. This is described in detail in Figure 4.



$$\text{Output at time 0} = W_0I_0$$
$$\text{Output at time 1} = W_1I_0 + W_0I_1$$
$$\text{Output at time 2} = W_2I_0 + W_1I_1 + W_0I_2$$
$$\text{Output at time 3} = W_3I_0 + W_2I_1 + W_1I_2 + W_0I_3$$
$$\vdots$$
$$\text{Output at time n} = W_nI_0 + W_{n-1}I_1 + W_{n-2}I_2 + \cdots + W_0I_n$$
$$= \sum_{t=0}^{n} W_{n-t}I_t$$

FIGURE 4. OUTPUT OF A LINEAR SYSTEM TO ARBITRARY INPUT DESCRIBING IT AS THE CONVOLUTION OF THE INPUT WITH THE IMPULSE RESPONSE OF THE SYSTEM

As can be clearly concluded from Figure 4, the numerical calculation for the convolution of two discretely sampled wavelets can be performed by time - reversing one of the wavelets and shifting it relative to the other wavelet, taking the sum of the products of corresponding elements after each shift. As an example, let's convolve wavelets a and b to obtain the output wavelet c.

$$a * b = c$$

$$\text{let } a = (a_0 \; a_1 \; a_2 \; a_3)$$
$$b = (b_0 \; b_1 \; b_2)$$

Let's time reverse wavelet b and shift it towards the right relative to wavelet a. The first non-zero product will be obtained when b_0 is aligned with a_0; i.e.,

$$\begin{array}{cccccc} & & (a_0 & a_1 & a_2 & a_3) \\ (b_2 & b_1 & b_0 & & & \end{array}$$
$$c_0 = a_0 b_0$$

The next value of the output is obtained by shifting the time-reversed version of wavelet b by one time unit, as follows:

$$\begin{array}{ccccc} & (a_0 & a_1 & a_2 & a_3) \\ (b_2 & b_1 & b_0) & & \end{array}$$
$$c_1 = a_1 b_0 + a_0 b_1$$

Similarly, subsequent values of the convolution are given by further shifts of b; i.e.,

$$\begin{array}{cccc} (a_0 & a_1 & a_2 & a_3) \\ (b_2 & b_1 & b_0) & \end{array}$$
$$c_2 = a_2 b_0 + a_1 b_1 + a_0 b_2$$

$$\begin{array}{ccccc} (a_0 & a_1 & a_2 & a_3) & \\ & (b_2 & b_1 & b_0) & \end{array}$$
$$c_3 = a_3 b_0 + a_2 b_1 + a_1 b_2$$

$$\begin{array}{cccccc} (a_0 & a_1 & a_2 & a_3) & & \\ & & (b_2 & b_1 & b_0) & \end{array}$$
$$c_4 = a_3 b_1 + a_2 b_2$$

$$\begin{array}{cccccc} (a_0 & a_1 & a_2 & a_3) & & \\ & & & (b_2 & b_1 & b_0) \end{array}$$
$$c_5 = a_3 b_2$$

At this point, we should introduce the concept of the *Z-transform.*

If we have a discrete wavelet a $=$ (a_0 a_1 a_2 a_3), we define the Z-transform of the wavelet a as a polynomial in Z (described as A(Z)), where the *coeffficients* of such polynomial are the amplitude values of the wavelet a, and the *powers of Z* represent the *time index* of their corresponding coefficient; that is,

$$\text{if} \quad a = (a_0 \; a_1 \; a_2 \; a_3) \text{ and } b = (b_0 \; b_1 \; b_2)$$
$$\text{then} \quad A(Z) = a_0 Z^0 + a_1 Z^1 + a_2 Z^2 + a_3 \cdot Z^3$$
$$= a_0 + a_1 Z + a_2 Z^2 + a_3 Z^3$$
$$\text{and} \quad B(Z) = b_0 + b_1 Z + b_2 Z^2$$

As a matter of interest, let's look at the multiplication of polynomials A(Z) and B(Z).

$$\begin{array}{rl} & a_0 + a_1 Z + a_2 Z^2 + a_3 Z^3 \\ \times & b_0 + b_1 Z + b_2 Z^2 \\ \hline & (a_0 b_0) + (a_1 b_0 + a_0 b_1) Z + (a_2 b_0 + a_1 b_1 + a_0 b_2) Z^2 \\ & + (a_3 b_0 + a_2 b_1 + a_1 b_2) Z^3 + (a_3 b_1 + a_2 b_2) Z^4 + (a_3 b_2) Z^5 \end{array}$$

The result is a new polynomial in Z which we may call C(Z).

Using the definition of the Z-transform, we can write this polynomial C(Z) in terms of the wavelet that it represents, by equating powers of Z to the equivalent time index of their coefficients; i.e.,

The coefficient associated with Z^0 (=1) is $a_0 b_0$;

then, $c_0 = a_0 b_0$

The coefficient of Z^1 is $a_1 b_0 + a_0 b_1$,

or $c_1 = a_1 b_0 + a_0 b_1$

Similarly,
$$c_2 = a_2 b_0 + a_1 b_1 + a_0 b_2$$
$$c_3 = a_3 b_0 + a_2 b_1 + a_1 b_2$$
$$c_4 = a_3 b_1 + a_2 b_2$$
$$c_5 = a_3 b_2$$

If we compare these results with those obtained by convolving wavelets a and b (see above) we will notice that both resultant wavelets c are identical. Thus we conclude that the convolution of two discrete wavelets is equivalent to the multiplication of their Z-transform; or "CONVOLUTION IN THE TIME-DOMAIN IS EQUIVALENT TO MULTIPLICATION IN THE Z-DOMAIN."

It can be proven that the Z-transform is the same as the Fourier transform when $Z = e^{-2\pi i f \Delta t}$, and so the Fourier transform can be considered as a special case of the Z-transform. Thus we can say that,

"CONVOLUTION IN THE TIME-DOMAIN IS EQUIVALENT TO MULTIPLICATION (OF THE AMPLITUDE SPECTRUM) IN THE FREQUENCY DOMAIN."

Another process similar to convolution which we should study at this point is *correlation* (denoted by the symbol ★).

"THE CORRELATION (OR CROSS CORRELATION) OF TWO WAVELETS IS A MEASURE OF THE SIMILARITY BETWEEN THE TWO WAVELETS AS A FUNCTION OF THE RELATIVE *TIME SHIFT* BETWEEN THEM."

The computational procedure is similar to convolution, except that in this case, the wavelets are *not* time-reversed relative to ane another. For example, the zero time-shift value or zeroth lag of the cross correlation between wavelets *a* and *b* would be:

$$\begin{matrix}(a_0 & a_1 & a_2 & a_3)\\ \updownarrow & \updownarrow & \updownarrow & \\ (b_0 & b_1 & b_2) & \end{matrix}$$

$$\Downarrow$$

$$\phi_0 = a_0b_0 + a_1b_1 + a_2b_2$$

Similarly,

$$\begin{matrix}(a_0 & a_1 & a_2 & a_3)\\ & \updownarrow & \updownarrow & \updownarrow \\ & (b_0 & b_1 & b_2)\end{matrix}$$

$$\Downarrow$$

$$\phi_1 = a_1b_0 + a_2b_1 + a_3b_2$$

$$\begin{matrix}(a_0 & a_1 & a_2 & a_3)\\ \updownarrow & \updownarrow & & \\ (b_0 & b_1 & b_2) & & \end{matrix}$$

$$\Downarrow$$

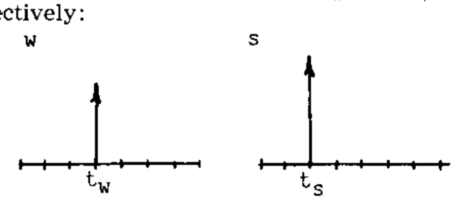
$$\phi_{-1} = a_0b_1 + a_1b_2$$

in general, $$\phi_\tau = \sum_{t=0}^{n} a_{t+\tau}\, b_t$$

where n is the length of the shortest wavelet.

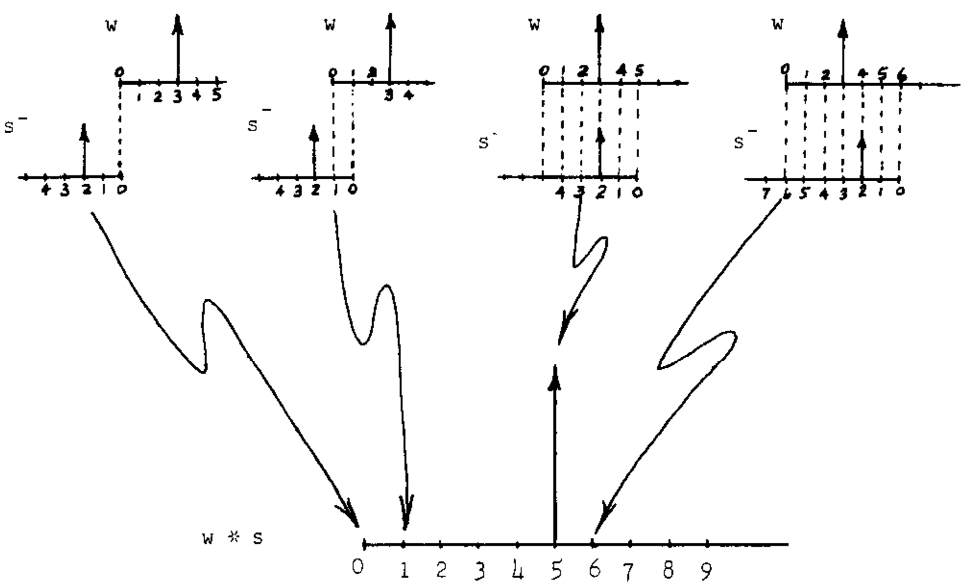
Most of the filtering techniques which we perform in seismic processing use some form of corelation, or more commonly *convolution,* and so it is extremely important to understand what these operations may do to our data.

Let's consider the two oversimplified wavelets, *w* and *s*, each consisting of only one non-zero value at times t_w and t_s respectively:



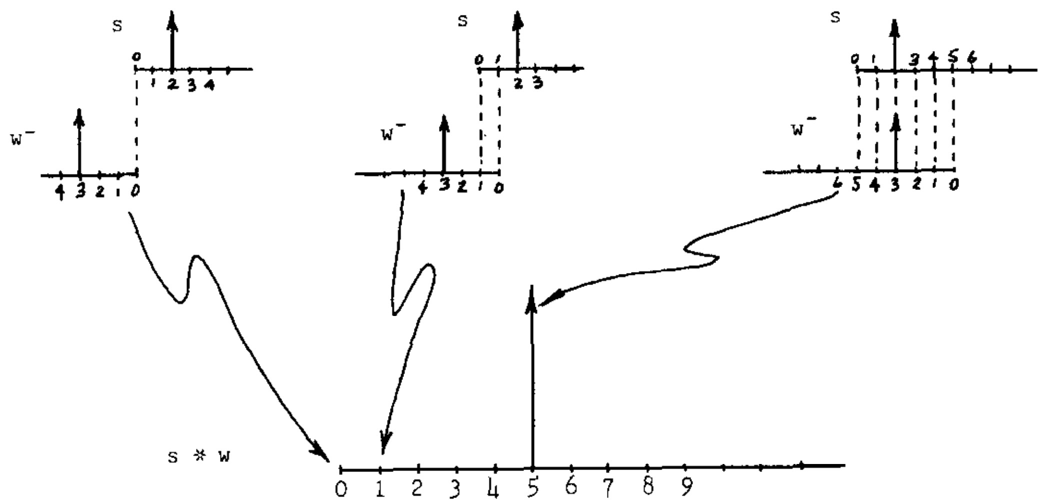
(Note that t is a representation of the *phase* of the wavelets since time shift is equivalent to phase shift.)

Let's take a numerical example as $t_w = 3$, $t_s = 2$, and see what happens when we convolve *w* with *s*. In this case, we will leave wavelet *w* stationary and time-reverse and move wavelet *s* past *w*.



We note that the only non-zero value of the convolution occurs at $t = t_w + t_s = 3 + 2 = 5$.

New, let's perform the same operation for the case of cross correlation, and see what happens when we correlate *w* with *s*.



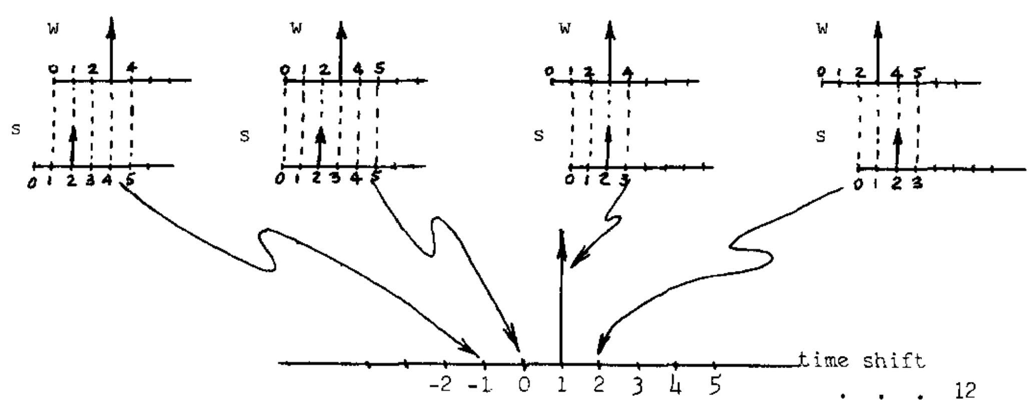
Let's look conversely at the convolution of *s* with *w;* keeping *s* stationary and time-reversing and moving *w* past *s*.

Again, we note that the only non-zero value occurs at $t = t_s + t_w = 2 + 3 = 5$.

From the above example, we can make the general conclusions:

In this case, we will leave the wavelet *w* stationary, and we will move *s* past *w*.

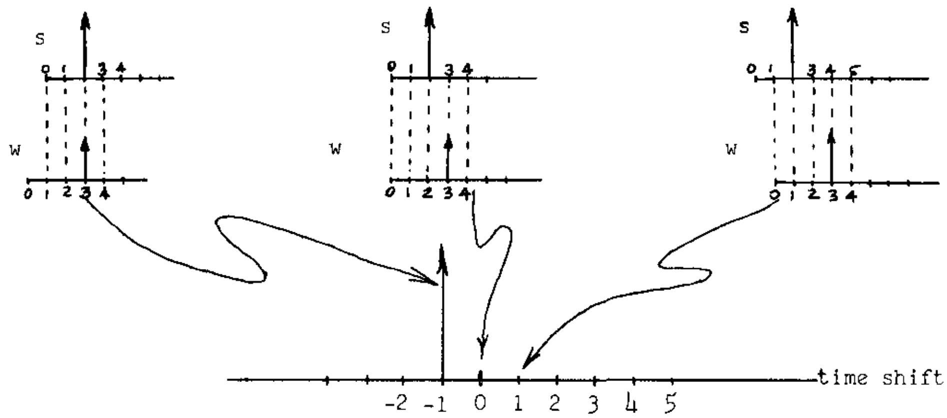
Note that all the values of the cross correlations are zero except for the value of the 1st lag (i.e., when *s* is shifted in time by + 1 units relative to *w*). Note that the non-zero value of the cross correlation occurs at $t = t_w - t_s = 3 - 2 = 1$.



1. "CONVOLUTION IS COMMUTATIVE; that is, w * s = s * w"
2. "CONVOLUTION IN THE TIME-DOMAIN IS EQUIVALENT TO *ADDITION* OF THE PHASE SPECTRA IN THE FREQUENCY DOMAIN".

Now let's look conversely at the correlation of *s* with *w*. In this case, we will keep *s* stationary and move *w* past *s*.

In this case, the values of the cross correlation are all zero except for lag -1 of the cross correlation; that is, when wavelet *w* is *delayed* by 1 time unit relative to *s*).



The non-zero value of the cross correlation occurs at $t = t_2 - t_1 = 2 - 3 = -1$.

Note that although the amplitude of the non-zero value of the cross correlation is the same for $w \star s$ as $s \star w$, its *phase* shift is different in the two cases. Therefore, we can conclude that in general,

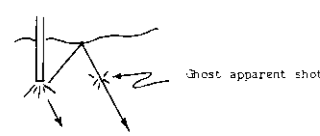
1. "CORRELATION IS NOT COMMUTATIVE"; that is, $w \star s \neq s \star w$.
2. "CORRELATION IN THE TIME-DOMAIN IS EQUIVALENT TO *SUBTRACTION* OF THE PHASE SPECTRA IN THE FREQUENCY-DOMAIN.' '

An exception to Rule 1 is the case of the autocorrelation of a wavelet, since obviously, $w \star w = w \star w$. Note that in this case, we have $t = t_1 - t_1 = 0$; that is, the autocorrelation of any wavelet is zero phase and therefore we cannot obtain from it any indication as to what the phase of the input wavelet was.

Deghosting

Let's look at the problem of removing a reverberation trend, or a ghost for example.

Some times we find that besides the energy that is transmitted from a seismic explosion deeper into the ground to be recorded at the geophones as seismic reflections, part of that energy goes back and bounces back either at the bottom of the weathering layer or at the surface[1]. In either case, we have energy travelling up from the source, bouncing back off a sharp velocity-difference interface and going back down; which means that at a time equal to the two-way travel time from the bottom of the shot hole to the shallower reflecting interface, we have a second apparent shot, and so we will find that at the different recorders, each primary reflector is represented, followed by a replica of that primary a few milli-seconds later, and normally reversed in polarity relative to the primary. We call this secondary event a 'ghost' of the preceding primary; caused by the aparent shot occurring a certain amount of time later (due to its longer travel time).



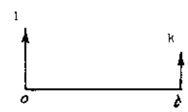
Of course, in the case that the reflecting interface is the surface, the delay is equal to twice the uphole time.

In the case that the bouncing interface is the bottom of the weathering layer, we don't know directly what the delay is, but we may be able to measure it by shifting the record relative to itself until we find each primary aligned with another event reversed in polarity; i.e., its ghost. The relative time shift between the records is then the delay or ghost time. This ghost lag can sometimes be measured by taking the autocorrelation of the trace, in which case we may see a strong trough in the auto-

[1]Although theoretically the ghost always occurs, unless we have a surface energy source, in many cases it is so small in amplitude that it is disregarded.

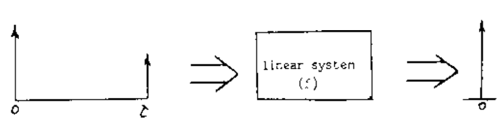
correlation, correlatable to the zeroth lag of the autocorrelation but inverted in polarity. That trough will be at a time lag equivalent to the ghost lag. However, in a great many cases, the ghost time lag is less than the period of the primary, and in these cases it could be extremely dificult to distinguish this ghost lag by either looking at the trace or its autocorrelation (this latter case will be generally handled by normal deconvolution).

So, in the case of a ghost, we have for each unit of primary energy arriving at the suface, a certain amount of ghost energy k ($|k| < 1$) arriving t milliseconds later, (where t represents the ghost time lag). That is, the energy generating system could be described as:



Of course, in the above model, only the first unit spike represents primary arrival, and so it would be desirable to design a filter that would transform such energy generating system into one composed of only the first unit spike, i.e., removing the ghost generating spike k at a time delay of t milliseconds.

We could pictorially describe such a process as the passage of the initial energy generating system through a linear system to obtain the desired result, as follows:



Obviously, the required filter f is the impulse response of the linear system that would perform the desired transformation. If we call the input to the linear system x, and desired output y, we can express the relationship between the input and the output of such a system as

$$x * f = y$$

(recall that the output of a linear system to any arbitrary input is given as the convolution of that arbitrary input with the impulse response of the system).

In terms of z-transforms, we can rewrite the above equations as

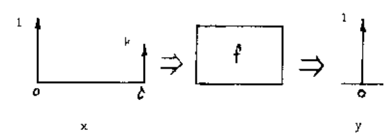
$$X(z) \cdot F(z) = Y(z)$$

(we will call this the basic equation of systems analysis).

That is, the z-transform of the desired filter is given by

$$F(z) = \frac{Y(z)}{X(z)}$$

or in our particular case,



$X(z) = 1 + kz^t$ $\qquad$ $Y(z) = 1$

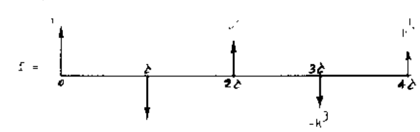
or,

$$F(z) = \frac{1}{1 + kz^t}$$

The result of this division is:

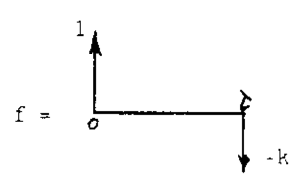
$$F(z) = 1 - kz^t + k^2 z^{2t} - k^3 z^{3t} + k^4 z^{4t} - k^5 z^{5t} + \dots$$

and thus,

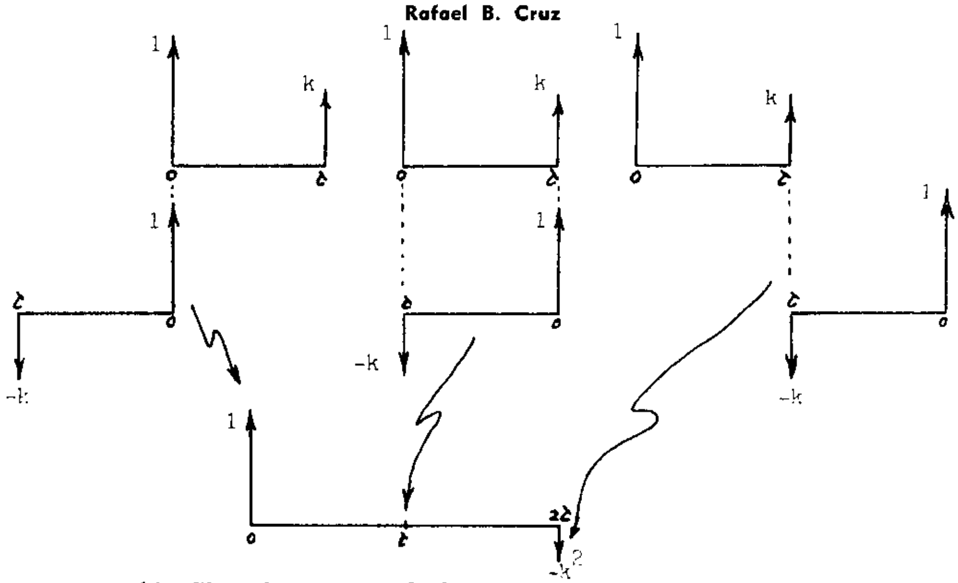


Note that the filter that we have obtained has infinite length. Of course, from a practical standpoint, this filter would have to be truncated after a certain point.

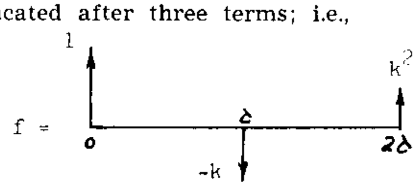
Let's suppose that such filter is truncated after the first two points; i.e., let



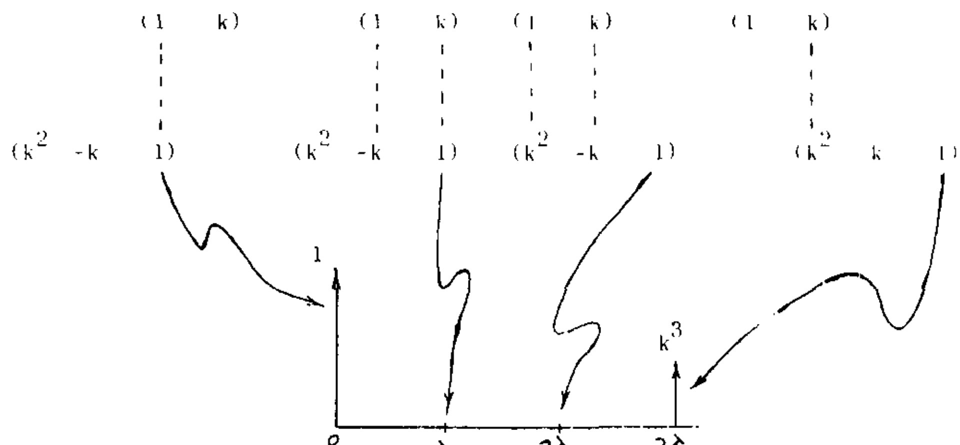
Applying such filter to the initial energy generating system,



As we see, this filter has removed the initial ghost generating system k at time t, but it has created a new spurious event at time $2t$ of amplitude $-k^2$. This spurious event is the result of truncating the filter after the first two terms. Of course, if $|k|<1$, the ampitude of this new undesirable event $(-k^2)$ is less than the amplitude of the initial ghost (k), and from that standpoint we could say that the filter has improved the data. A more accurate result could be obtained by using a longer filter. Suppose for example, that the filter is truncated after three terms; i.e.,



Now, convolving our initial energy generating system with this filter

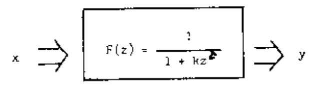


In this case, the truncation error associated with the filtering process is of the order of k^3.

As can be seen, the truncation error can be made smaller by considering longer filters, since it is the last term of the truncated filter that generates the error, and the filter coefficients are deceasing in amplitude with time. Such a type of filter is called a *stable* filter (we will expand on this concept of filter stability later on).

Let's attempt to view this deghosting process from a somewhat different standpoint, that is, let's analyze the relationship between any input data point to the linear system with transfer function

$$F(z) = \frac{1}{1 + kz^{t}}$$



in terms of z-transforms,

$$X(z) \cdot F(z) = Y(z)$$

or

$$X(z) \cdot \frac{1}{1 + kz^{t}} = Y(z)$$

$$X(z) = (1 + kz^{t})\, Y(z)$$

$$X(z) = Y(z) + kz^{t}\, Y(z)$$

or, at any arbitrary time t,

$$x_t = y_t + ky_{t-t}$$

(Note that the multiplication of Y(z) by z^t results in time delay of *t* units. This can be clearly seen if we recall that multiplication in the z-domain is equivalent to convolution in the time-domain, and so

$$Kz^t \,.\, Y(z)$$

can be viewed as the convolution of the output data string *y* with a wavelet of the form

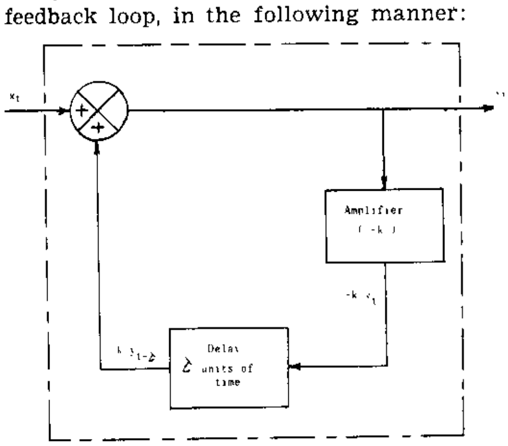
$$(\underset{0}{0}\ 0\ .\ .\ .\ 0\ \underset{t}{k})\ ).$$

Rewriting the above equation in terms of y_t, we obtain the following recursive relation:

$$y_t = x_t - ky_{t-t}$$

The foregoing equation describes the output at time t as a function of the input at the present time t, and a past value of the output, at time t-*t*.

The above type of filtering process is called *feedback* filtering, or is also sometimes called recursive or regressive filtering. The particular filter above can be easily described in terms of an anolog feedback loop, in the following manner:



ANALOG DEGHOSTING FILTER

In terms of digital processing, we can best describe the application of the feedback deghosting filter by looking at a particular example. Suppose that $t = 10$ ms.

then,

$$y_0 = x_0 - ky_{0-10} = x_0$$

$$y_1 = x_1 - ky_{-9} = x_1$$

$$\vdots$$

$$y_9 = x_9 - ky_{-1} = x_9$$

$$y_{10} = x_{10} - ky_0 = x_{10} - kx_0$$

$$y_{11} = x_{11} - ky_1 = x_{11} - kx_1$$

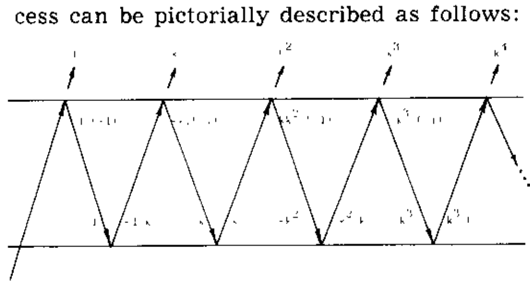
(if the data start at time zero, we can assume all the values before time zero are $= 0$).

Note that for every output data point, only *one* multiplication and *one* addition is required, and thus, this feedback process can be performed very quickly on a digital computer.

DEREVERBERATION

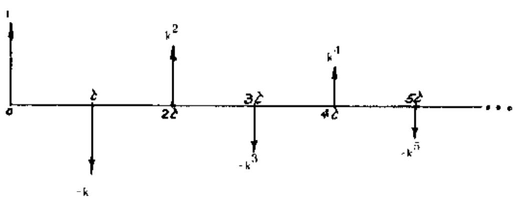
Let's now consider the case of water reverberation in marine seismic data. For all practical purposes, the water-air interface can be considered to have a reflection coefficient equal to *-1*; i.e., the surface of the water acts as a perfect reflector. In many cases also, the bottom of the water represents an interface with a high reflection coeffiicent. In these cases, the water layers acts as an energy trap causing multiple events to occur, with period equal to the two-way travel time through the water layer.

If we first view the reverberation as occurring only at the receiver, we have, for each unit of primary energy arriving at the surface, certain amounts of reverberation arriving at intervals of *t* milliseconds from the primary arrival. The process can be pictorially described as follows:

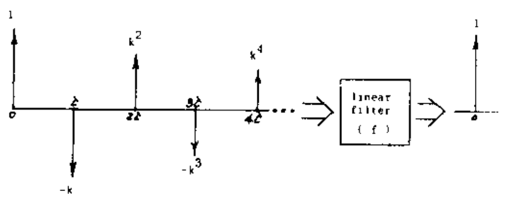


That is, the unit of primary energy arriving at the surface is reflected back into the water with reflection coefficient of *-1*, and thus the amount of energy going back into the water layer is 1 (-1) = -1. This energy, upon arriving at the bottom of the water layer, is reflected back up with reflection coefficient equal to k, and so the energy reaching the water surface as a first order reverberation is equal to -1 (k) = -k. In the same manner, we can conclude that the energy of the second order reverberation is k^2, and that for third and higher order reverberations, the corresponding energy arrivals are $-k^3$, k^4, $-k^5$, k^6, etc.

Thus, the energy generating system in the case of simple water reverberation can be described as,



Obviously, all of those spikes except the first one represent reverberations, and so it would be desirable to design a filter that would transform such energy generating system into one composed of only the first unit spike. That is, we wish to design a filter such that



i.e., $x * f = y$

or $X(z) \cdot F(z) = Y(z)$

and since $X(z) = 1 - kz^{t} + k^2z^{2t} - k^3z^{3t} + k^4z^{4t} - k^5z^{5t} + \ldots$

$Y(z) = 1$

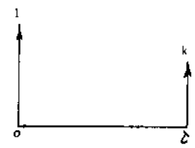
then, $$F(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - kz^{t} + k^2z^{2t} - k^3z^{3t} + k^4z^{4t} - \ldots}$$

Now, if we recall that the denominator of the above expression is the same as the inverse of the ghost generating mechanism, we can rewrite it as,

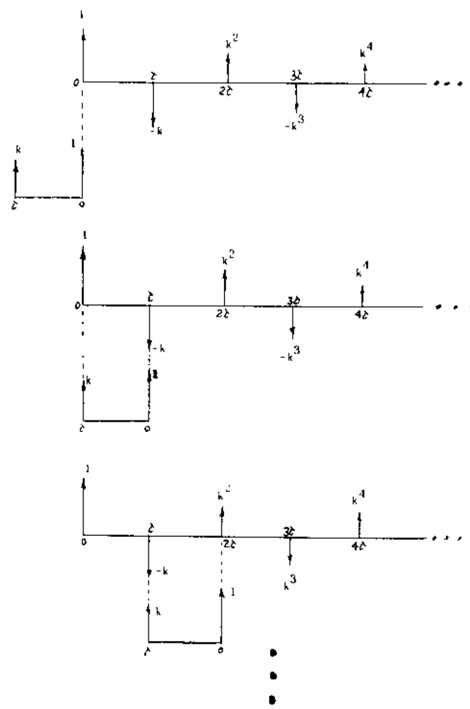
$$F(z) = \frac{1}{\dfrac{1}{1 + kz^{t}}}$$

or, $F(z) = 1 + kz^{t}$

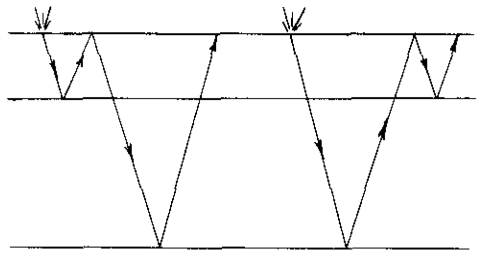
and the simple dereverberation filter would then be:



The performance of this filter can be evaluated by convolving the reverberation generating system with it, as follows

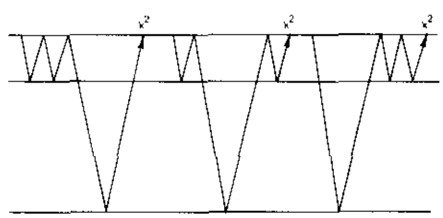


However, reverberation occurs also at the energy source, as well as at the receivers. That means that a first order reverberation, arriving *t* milliseconds later than the primary arrival (note that *t* is simply the two-way travel time through the water layer) could have either occurred before the energy went deeper into the ground (at the source), or after the energy emerged back into the water layer, as follows



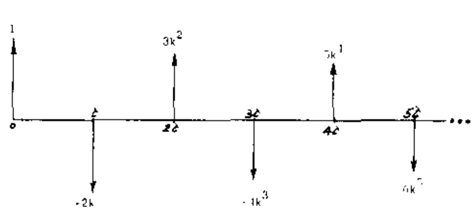
If we assume that the water depth is constant between the source and the receiver, both of the above reverberations will arrive *in phase* t milliseconds later than the primary event, and thus for each unit of primary energy arriving at the surface, we will have an amount of energy equal to $-2k$ arriving as the first order reverberation (i.e., the sum of the amplitudes of the two types of first order reverberation occurring, or (-k) + (-k)).

Similarly, three different types of second order reverberation will occur, each with amplitude k^2, and thus the resultant second order reverberation will be composed of the in-phase addition of those three reverberations, with a combined amplitude equal to $3k^2$, and arriving at a time of $2t$ milliseconds later than the primary

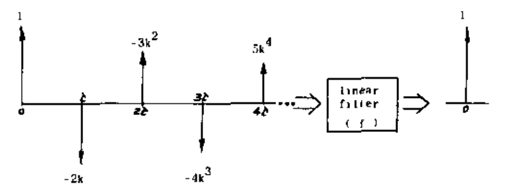


SECOND ORDER REVERBERATIONS

In the same manner, we can show that there will be four different third-order reverberations ,five fourth-order, six fifth-order, and so on; and therefore, the complete water reverberation mechanism will be,



Of which, again, only the first unit spike represents primary arrival, and so a dereverberation filter should be one that will transform the above energy generating system into only the initial unit spike.



or, $x * f = y$

in terms of z-transforms,

$$X(z) \cdot F(z) = Y(z)$$

$$X(z) = 1 - 2kz^{t} + 3k^2z^{2t} = 4k^3z^{3t} + 5k^4z^{4t} \ldots$$

$$Y(z) = 1$$

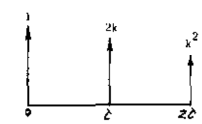
$$F(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - 2kz^{t} + 3k^2z^{2t} - 4k^3z^{3t} + 5k^4z^{4t} \ldots}$$

the result of the above division is:

$$F(z) = 1 + 2kz^{t} + k^2z^{2t}$$

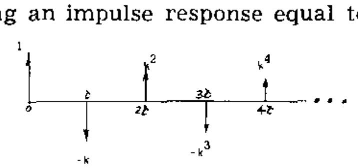
or,

f =

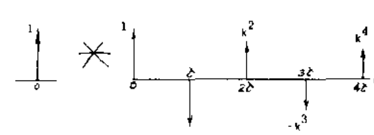


(this filter is normally called the 3-point Backus filter)

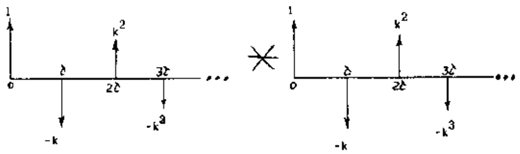
The complete water reverberation could be analyzed from a different standpoint, if we view the water layer as a linear filter having an impulse response equal to:



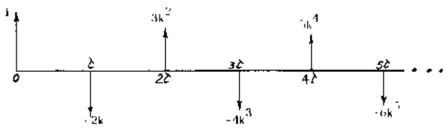
Then, the passage of a unit of primary energy through the water layer, could be described as the convolution of that unit energy with the impulse response of the water layer; i.e.,



Now, as that energy emerges from the earth, it will pass through the water layer again, and thus we will have the data being convolved again with the impulse response of the water layer,



The result of this convolution we can clearly see as



which is the same energy generating system as we obtained before.

Going back to our filter design, we have,

$$F(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - 2kz^{t} + 3k^2z^{2t} - 4k^3z^{3t} + 5k^4z^{4t} \ldots}$$

which we can now rewrite as

$$F(z) = \frac{1}{(1 - kz^{t} + k^2z^{2t} - k^3z^{3t} + \ldots)\cdot(1 - kz^{t} + k^2z^{2t} - k^3z^{3t} + \ldots)}$$

or,

$$F(z) = \frac{1}{(1 - kz^{t} + k^2z^{2t} - k^3z^{3t} + \ldots)^2}$$

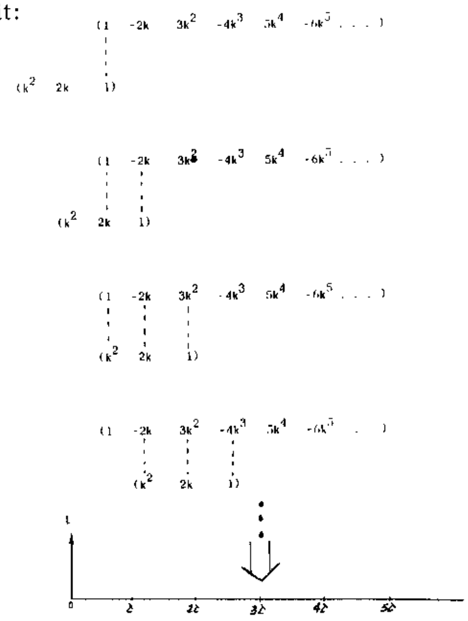
i.e.,

$$F(z) = (1 + kz^{t})^2$$

or $$F(z) = 1 + 2kz^{t} + k^2z^{2t}$$

which is the same filter which we obtained before.

The performance of this filter can be seen by convolving the complete water reverberation energy generating system with it:



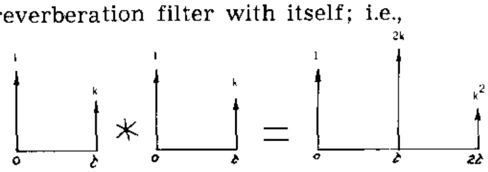
From the above convolution,

$$y_0 = 1 \cdot 1 = 1$$
$$y_{t} = 1 \cdot (-2k) + 2k(1) = -2k + 2k = 0$$
$$y_{2t} = 1 \cdot (3k^2) + 2k(-2k) + k^2(1) = 3k^2 - 4k^2 + k^2 = 0$$
$$y_{3t} = 1 \cdot (-4k^3) + 2k(3k^2) + k^2(-2k) = -4k^3 + 6k^3 - 2k^3 = 0$$
$$y_{4t} = 1 \cdot (5k^4) + 2k(-4k^3) + k^2(3k^2) = 5k^4 - 8k^4 + 3k^4 = 0$$

.
.
.

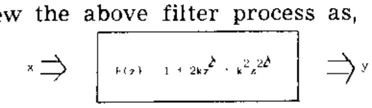
So, as we see, the result of the convolution has eliminated the reverberation mechanism, leaving only the primary arrival.

It is interesting to notice that the above filter is the convolution of the simple dereverberation filter with itself; i.e.,



and so, the same result could have been obtained by convolving the data *twice* with the simple dereverberation filter.

In terms of any arbitrary data input, we could view the above filter process as,



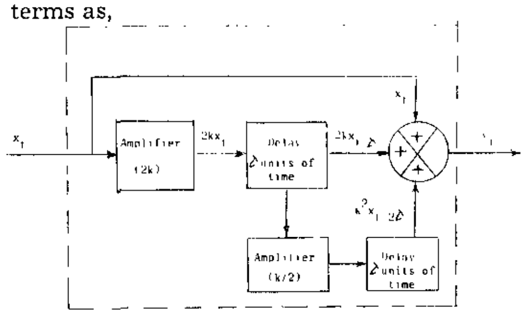
or,

$$X(z) \cdot F(z) = Y(z)$$
$$X(z) \cdot (1 + 2kz^{t} + k^2z^{2t}) = Y(z)$$
$$X(z) + 2kz^{t}X(z) + k^2z^2\,Z(z) = Y(z)$$

at time t:

$$x_t + 2kx_{t-t} + k^2x_{t-2t} = y$$

we can then describe this filter in analog terms as,



ANALOG DEREVERBERATION FILTER

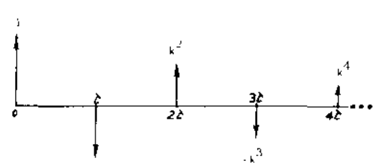
However, we have to keep in mind that the previously designed filters are of a deterministic type; i.e., they assume that we know what the appropriate values for k and t are.

Of these two parameters, the most sensitive is t since, if we have an incorrect value for it, the filtering process will be performed adding the wrong values of the input data, and consequently, our results will be very poor. On the other hand, an approximate value for k (if we use the correct t) will do a good job of attenuating the undesirable events, and if the error in estimating k is less than 100% of the actual value, the results of filtering will show an improvement over the input data.

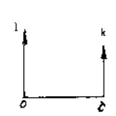
Note that if our estimate for k is too low, the results will show some residual multiple energy, whereas, if our estimate for k is too high, the remaining multiple energy will appear inverted in polarity relative to the way it was in the input data.

In any event, whether we use the above filters to process the data or not, they serve to give us some insight into what the general characteristics of the filters would be, and the effectiveness of their application. With that in mind, we can then proceed to pick appropriate parameters for a statistical filter design that would effectively process the data.

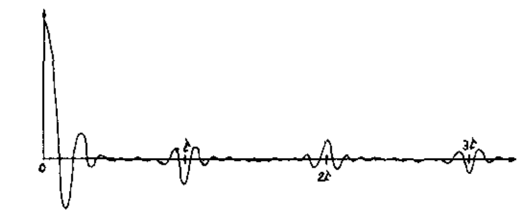
For example, let's consider again, the model for a simple water reverberation



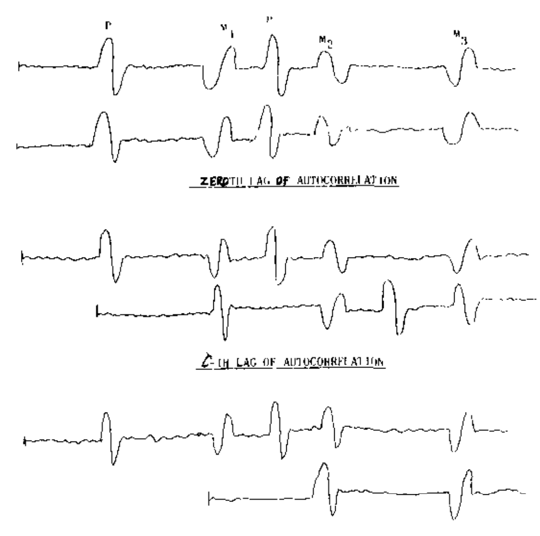
for which we found that the optimum filter was



Note that the above model is the same as the one describing any surface multiples. In general, the value of t, or multiple period will be large enough that if we display the autocorrelation of the input trace we will clearly see a high amplitude trough in the autocorrelation (if high order multiples are included within the autocorrelation window) correlatable to the zeroth lag of that autocorrelation, and occurring at a time-lag of t milliseconds; i.e.,



Note that the same multiple periodicity shows up in the autocorrelation (i.e., at times t, $2t$, $3t$, etc.). We can see this clearly if we view how the autocorrelation is computed:



This process will give us a very accurate value for t for the trace whose autocorrelation we have displayed, and we can obtain an estimate for k by comparing the tth lag of the autocorrelation to the zeroth lag.

However, the values for t will normally change from trace to trace and shot to shot, and so it becomes a cumbersome process to estimate the actual value of t for every trace. If, on the other hand, we make a guess for t that would be *less* than the actual prediciton time, and then we design a long enough least squares "prediction" filter such that the actual period will fall between the guessed value of t and that value plus the filter length, we will find that such filter will contain a large spike at the actual prediction time, thus monitoring t as it changes from trace to trace.

In the past the application of conventional predictive deconvolution techniques has been relatively unsuccessful in handling the characteristic non-periodicity of the water bottom multiples evident in long offset traces in marine data. The problem can be resolved by first correcting the seismic records with a constant water velocity. This procedure has the effect of "flattening" all the water bottom multiples, thus creating periodicity on all the traces. Conventional predictive deconvolution may then be successfully applied to attenuate the water bottom multiples. The data is then "un-NMO" corrected by removing the delta-t applied by water velocity, thus recreating the original seismic records without the water bottom multiples.