Lattice Realization

Here we will study another FIR filter structure called the LATTICE FILTER or lattice realization.

Lattice filters are extensively used in speech processing and in the implementation of adaptive filters.

Let us consider a sequence of FIR filters with system functions:

\[ H_m[z] = A_m[z] \quad m = 0,1,2,\ldots,(M-1) \]

where \( A_m[z] \) is the polynomial:

\[ A_m[z] = 1 + \sum_{k=1}^{m} \alpha_m(k) z^{-k} \quad m \geq 1 \]

and \( A_0[z] = 1 \)

The unit impulse response of the \( m \)th filter is \( h_m(0) = 1 \) and \( h_m(k) = \alpha_m(k) \) for \( k = 1,2,\ldots,m \).

It is desirable to view FIR filters as linear predictors since the input sequence \( x[n-1], x[n-2], \ldots, x[n-m] \) can be used to predict the value of the signal \( x[n] \).

The linearly predicted value of \( x[n] \) equals:

\[ \hat{x}[n] = -\sum_{k=1}^{m} \alpha_m[k] x[n-k] \]

where \( \{ - \alpha_m[k] \} \) represent the prediction coefficients.

The output sequence \( y[n] \) may be expressed as:

\[ y[n] = x[n] - \hat{x}[n] = x[n] + \sum_{k=1}^{m} \alpha_m[k] x[n-k] \quad (1) \]

Hence the FIR filter output given by equation (1) may be interpreted as the error between the true signal value \( x[n] \) and the predicted value \( \hat{x}[n] \).

Two direct form realization of the FIR prediction filter are shown below:
Suppose that we have a filter for which \( m = 1 \). Then the output of this filter is:

\[
y[n] = x[n] + \alpha_1[1]x[n - 1]
\]  

(2)

This output can also be obtained from the first order or single-stage lattice filter shown below:
\[ f_0[n] = g_0[n] = x[n] \]
\[ f_1[n] = f_0[n] + k_1 g_0[n-1] = x[n] + k_1 x[n-1] \]
\[ g_1[n] = k_1 f_0[n] + g_0[n-1] = k_1 x[n] + x[n-1] \]
If we select \( k_1 = \alpha_1 \) the \( f_1[n] \) will be equal to equation (2)

Parameter \( k_1 \) in the lattice filter is called the “reflection coefficient”.

Now consider an FIR filter for which \( m=2 \).
In this case the output from a direct-form structure is:
\[ y[n] = x[n] + \alpha_2[1] x[n-1] + \alpha_2[2] x[n-2] = x[n] + \sum_{k=1}^{2} \alpha[k] x[n-k] \]
\[ y[n] = x[n] + \alpha_2[1] x[n-1] + \alpha_2[2] x[n-2] \quad (3) \]

Two stage Lattice Filter

Output from the first stage is:
\[ f_1[n] = x[n] + k_1 x[n-1] \]
\[ g_1[n] = k_1 x[n] + x[n-1] \]

the output from the second stage is:
\[ f_2[n] = f_1[n] + k_2 g_1[n-1] \]
\[ g_2[n] = f_1[n] k_2 + g_1[n-1] \]

or equivalently using the results of stage-1
\[ f_2[n] = x[n] + k_1 x[n-1] + k_1 k_2 x[n-1] + k_2 x[n-2] \]
\[ = x[n] + k_1 (1 + k_2) x[n-1] + k_2 x[n-2] \quad (4) \]
Note here that equation (4) is identical to equation (3) if we equate:

$$\alpha_2[2] = k_2, \quad \alpha_2[1] = k_1(1 + k_2)$$

Or equivalently:

$$k_2 = \alpha_2[2]$$

$$k_1 = \frac{\alpha_2[1]}{1 + \alpha_2[2]}$$

hence the reflection coefficients $k_1$ and $k_2$ of the lattice filter can be obtained from the coefficients $\{\alpha_m[k]\}$ of the direct form realization.

By continuing this process one can easily demonstrate by induction the equivalence between an $m$th order direct-form FIR filter and an $m$-stage lattice filter. The lattice filter is generally described by the following set of order-recursive equations.

$$f_0[n] = g_0[n] = x[n]$$

$$f_m[n] = f_{m-1}[n] + k_m g_{m-1}[n - 1] \quad m = 1, 2, \ldots, M - 1$$

$$g_m[n] = k_m f_{m-1}[n] + g_{m-1}[n - 1] \quad m = 1, 2, \ldots, M - 1$$

The output of the $(M-1)$th stage corresponds to output of an $(M-1)$ order FIR filter.

i.e. $y[n] = f_{M-1}[n]$

As a consequence of the equivalence between an FIR filter and a lattice filter the output $f_m[n]$ of an $m$-stage lattice filter can be written as:

$$f_m[n] = \sum_{k=0}^{m} \alpha_m[k] x[n-k] \quad \text{where, } \alpha_m[0] = 1$$

(5)

Since eq.(5) is a convolution sum it follows that the Z-transform is a simple multiplication.

$$F_m[z] = A_m[z] X[z]$$

or equivalently

$A_m[z] = \frac{F_m[z]}{X[z]} = \frac{F_m[z]}{F_0[z]}$
The second output $g_m[n]$ of the lattice can also be expressed as a convolution sum by using an other set of coefficients $\{\beta_m(k)\}$.

The output $g_m[n]$ from an m-stage lattice filter may be expressed by the convolution sum of the form:

$$g_m[n] = \sum_{k=0}^{m} \beta_m[k] x[n-k]$$

where, $\beta_m[0] = 1$ \hspace{1cm} (6)

where the filter coefficients $\{\beta_m(k)\}$ are associated with a filter that produces $f_m[n] = y[n]$ but operates in reverse order.

Supposing $x[n], x[n-1], \ldots, x[n-m+1]$ is used to linearly predict the signal value $x[n-m]$. The predicted value equals:

$$\hat{x}[n-m] = -\sum_{k=0}^{m-1} \beta_m[k] x[n-k]$$

where the coefficients $\beta_m(k)$ in the prediction filter are simply the coefficients $\{\alpha_m[k]\}$ taken in reverse order.

$$\beta_m[k] = \alpha_m[m-k] \hspace{1cm} k = 0,1,\ldots,m$$

Now if we consider the same in Z-domain, the transform of eq. (6) becomes:

$$G_m[z] = B_m[z]X[z]$$

$$\therefore \hspace{0.5cm} B_m[z] = \frac{G_m[z]}{X[z]}$$

here $b_m[z]$ represents the system function of the FIR filter with coefficients $\beta_m(k)$.

$$B_m[z] = \sum_{k=0}^{m} \beta_m[k] z^{-k}$$

Since $\beta_m(k) = \alpha_m(m-k)$

$$B_m[z] = \sum_{k=0}^{m} \alpha_m[m-k] z^{-k}$$

$$= \sum_{l=0}^{m} \alpha_m[l] z^{-l-m}$$

$$= z^{-m} \sum_{l=0}^{m} \alpha_m[l] z^{l-m}$$

$$= z^{-m} A_m[z^{-1}]$$
This implies that zeros of the FIR filter with system function \( B_m[z] \) are simply the reciprocal of the zeros of \( A_m[z] \). Hence \( B_m[z] \) is called the reciprocal polynomial of \( A_m[z] \).

Now that this relationship is shown let’s get back to recursive lattice equations and transfer them to z-domain.

\[
F_o[z] = G_o[z] = X[z]
\]
\[
F_m[z] = F_{m-1}[z] + K_m z^{-1} G_{m-1}[z]
\quad m = 1, 2, \ldots, m - 1
\]
\[
G_m[z] = K_m F_{m-1}[z] + z^{-1} G_{m-1}[z]
\quad m = 1, 2, \ldots, m - 1
\]

If we divide each equation by \( X[z] \)

\[
A_o[z] = B_o[z] = 1
\]
\[
A_m[z] = A_{m-1}[z] + K_m z^{-1} B_{m-1}[z]
\quad m = 1, 2, \ldots, m - 1
\]
\[
B_m[z] = K_m A_{m-1}[z] + z^{-1} B_{m-1}[z]
\quad m = 1, 2, \ldots, m - 1
\]

Hence the lattice stage is described in the z-domain by the matrix equation:

\[
\begin{bmatrix}
A_m[z] \\
B_m[z]
\end{bmatrix}
= \begin{bmatrix}
1 & K_m \\
K_m & 1
\end{bmatrix}
\begin{bmatrix}
A_{m-1}[z] \\
z^{-1} B_{m-1}[z]
\end{bmatrix}
\]

The lattice coefficients \( \{K_i\} \) can be converted to direct form coefficients \( \{\alpha_m[k]\} \) as shown below:

\[
A_o[z] = B_o[z] = 1
\]
\[
A_m[z] = A_{m-1}[z] + K_m z^{-1} B_{m-1}[z]
\quad m = 1, 2, \ldots, m - 1
\]
\[
B_m[z] = z^{-m} A_m[z^{-1}]
\quad m = 1, 2, \ldots, m - 1
\]

Solution is obtained recursively beginning with \( m = 1 \). Thus we obtain a sequence of \( (M-1) \) FIR filters one for each value of \( m \).

**Example:**

Given a three-stage lattice filter with coefficients \( K_1 = 1/4 \), \( K_2 = 1/2 \), \( K_3 = 1/3 \), determine the FIR filter coefficients for the direct form structure.

We can solve this problem recursively. Let us begin with \( m = 1 \). Thus we have:

\[
A_1[z] = A_o[z] + K_1 z^{-1} B_o[z]
\]
\[
= 1 + K_1 z^{-1} = 1 + \frac{1}{4} z^{-1}
\]
Hence the coefficients of an FIR filter corresponding to the single-stage lattice are:
\[ \alpha_1(0) = 1, \alpha_1(1) = K_1 = \frac{1}{4} \]. Since \( B_m[z] \) is the reverse polynomial of \( A_m[z] \) we have:
\[ B_1[z] = \frac{1}{4} + z^{-1} \]

Next we add the second stage to the lattice. For \( m=2 \)
\[
A_2[z] = A_1[z] + K_2z^{-1}B_1[z]
= 1 + \frac{3}{8}z^{-1} + \frac{1}{2}z^{-2}
\]

Hence the FIR filter parameters corresponding to the two-stage lattice are
\[ \alpha_2(0) = 1, \alpha_2(1) = \frac{3}{8}, \alpha_2(2) = \frac{1}{2} \]. Also
\[ B_2[z] = \frac{1}{2} + \frac{3}{8}z^{-1} + z^{-2} \]

Finally the addition of the third stage to the lattice results in the polynomial
\[
A_3[z] = A_2[z] + K_3z^{-1}B_2[z]
= 1 + \frac{13}{24}z^{-1} + \frac{5}{8}z^{-2} + \frac{1}{3}z^{-3}
\]

Consequently, the desired direct-form FIR filter is characterized by the coefficients
\[ \alpha_3(0) = 1, \quad \alpha_3(1) = \frac{13}{24}, \quad \alpha_3(2) = \frac{5}{8}, \quad \alpha_3(3) = \frac{1}{3} \]

MATLAB for calculating reflection coefficients
M-functions:
poly2rc
rc2poly

\[
\begin{align*}
\text{k} &= [1/4 \hspace{1cm} 1/2 \hspace{1cm} 1/3 ] \\
\text{k} &= \hspace{1cm} 0.2500 \hspace{1cm} 0.5000 \hspace{1cm} 0.3333
\end{align*}
\]

>> rc2poly(k)
ans =
    1.0000    0.5417    0.6250    0.3333

>>

OR

>> a=[1 13/24 5/8 1/3]
a =
    1.0000    0.5417    0.6250    0.3333

>> poly2rc(a)
ans =
    0.2500
    0.5000
    0.3333

>>