Carnegie Mellon DSP-I Electrical & Computer ENGINEERING

Digital Signal Processing I (18-791)
Fall Semester, 2005

THE FREQUENCY SAMPLING FIR FILTER IMPLEMENTATION

Issued: 11/04/05

Note: This handout is a copy of the material in the supplementary text *Digital Signal Processing Using* MATLAB by V. K. Ingle and J. G. Proakis (Brooks/Cole, 2000) on the frequency-sampling implementation of FIR filters.

FREQUENCY SAMPLING FORM

In this form we use the fact that the system function $H(z)$ of an FIR filter can be reconstructed from its samples on the unit circle. From our discussions on the DFT in Chapter 5 we recall that these samples are in fact the M-point DFT values $\{H(k), \quad 0 \le k \le M-1\}$ of the M-point impulse response $h(n)$. Therefore we have

$$\begin{aligned} H(z) &= \mathcal{Z}[h(n)] \\ &= \mathcal{Z}[\text{IDFT}\{H(k)\}] \end{aligned}$$

Using this procedure, we obtain [see (5.17) on page 127]

$$H(z) = \left(\frac{1-z^{-M}}{M}\right) \sum_{k=0}^{M-1} \frac{H(k)}{1 - W_M^{-k} z^{-1}} \tag{6.12}$$

This shows that the DFT $H(k)$, rather than the impulse response $h(n)$ (or the difference equation), is used in this structure. It is also interesting to note that the FIR filter described by (6.12) has a recursive form similar to an IIR filter because (6.12) contains both poles and zeros. The resulting filter is an FIR filter since the poles at W_M^{-k} are canceled by the roots of

$$1 - z^{-M} = 0$$

The system function in (6.12) leads to a parallel structure as shown in Figure 6.15 for $M = 4$.

One problem with the structure in Figure 6.15 is that it requires a complex arithmetic implementation. Since an FIR filter is almost always a real-valued filter, it is possible to obtain an alternate realization in which only real arithmetic is used. This realization is derived using the symmetry properties of the DFT and the W_M^{-k} factor. Then (6.12) can be expressed

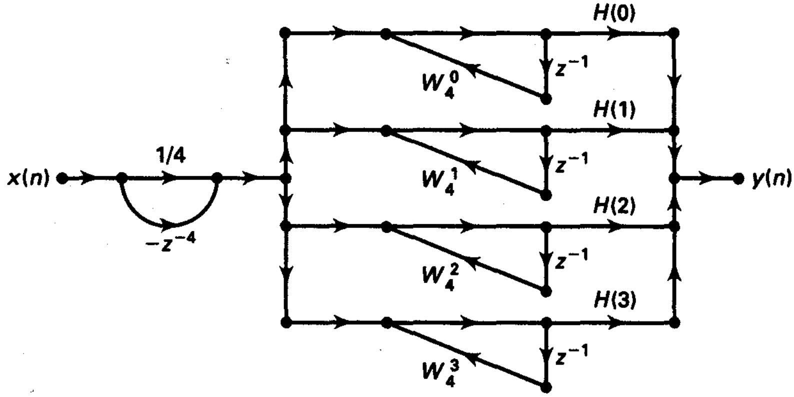

FIGURE 6.15 *Frequency sampling structure for $M = 4$*

as (see Problem 6.10)

$$H(z) = \frac{1 - z^{-M}}{M} \left\{ \sum_{k=1}^{L} 2 \left| H(k) \right| H_k(z) + \frac{H(0)}{1 - z^{-1}} + \frac{H(M/2)}{1 + z^{-1}} \right\} \quad \textbf{(6.13)}$$

where $L = \frac{M-1}{2}$ for M odd, $L = \frac{M}{2} - 1$ for M even, and $\{H_k(z), \quad k = 1, \ldots, L\}$ are second-order sections given by

$$H_k(z) = \frac{\cos\left[\angle H(k)\right] - z^{-1} \cos\left[\angle H(k) - \frac{2\pi k}{M}\right]}{1 - 2z^{-1} \cos\left(\frac{2\pi k}{M}\right) + z^{-2}} \quad \textbf{(6.14)}$$

Note that the DFT samples $H(0)$ and $H(M/2)$ are real-valued and that the third term on the right-hand side of (6.13) is absent if M is odd. Using (6.13) and (6.14), we show a frequency sampling structure in Figure 6.16 for $M = 4$ containing real coefficients.

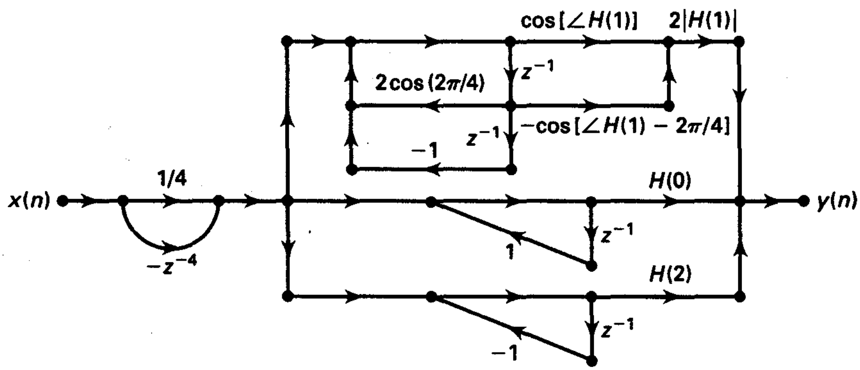

FIGURE 6.16 *Frequency sampling structure for $M = 4$ with real coefficients*

MATLAB IMPLEMENTATION

Given the impulse response $h(n)$ or the DFT $H(k)$, we have to determine the coefficients in (6.13) and (6.14). The following MATLAB function, dir2fs, converts a direct form ($h(n)$ values) to the frequency sampling form by directly implementing (6.13) and (6.14).

```
function [C,B,A] = dir2fs(h)
% Direct form to Frequency Sampling form conversion
% -----------------------------------------------
% [C,B,A] = dir2fs(h)
% C = Row vector containing gains for parallel sections
% B = Matrix containing numerator coefficients arranged in rows
% A = Matrix containing denominator coefficients arranged in rows
% h = impulse response vector of an FIR filter
%
M = length(h);
H = fft(h,M);
magH = abs(H); phaH = angle(H)';
% check even or odd M
if (M == 2*floor(M/2))
      L = M/2-1;    % M is even
     A1 = [1,-1,0;1,1,0];
     C1 = [real(H(1)),real(H(L+2))];
else
      L = (M-1)/2; % M is odd
     A1 = [1,-1,0];
     C1 = [real(H(1))];
end
k = [1:L]';
% initialize B and A arrays
B = zeros(L,2); A = ones(L,3);
% compute denominator coefficients
A(1:L,2) = -2*cos(2*pi*k/M); A = [A;A1];
% compute numerator coefficients
B(1:L,1) = cos(phaH(2:L+1));
B(1:L,2) = -cos(phaH(2:L+1)-(2*pi*k/M));
% compute gain coefficients
C = [2*magH(2:L+1),C1]';
```

In the above function the impulse response values are supplied through the h array. After conversion, the C array contains the gain values for each parallel section. The gain values for the second-order parallel sections are given first, followed by $H(0)$ and $H(M/2)$ (if M is even). The B matrix contains the numerator coefficients, which are arranged in length-2 row vectors for each second-order section. The A matrix contains the denominator coefficients, which are arranged in length-3 row vectors for the second-order sections corresponding to those in B, followed by the coefficients for the first-order sections.

A practical problem with the structure in Figure 6.16 is that it has poles on the unit circle, which makes this filter critically unstable. If the filter is not excited by one of the pole frequencies, then the output is bounded. We can avoid this problem by sampling $H(z)$ on a circle $|z| = r$, where the radius r is very close to one but is less than one (e.g., $r = 0.99$), which results in

$$H(z) = \frac{1 - r^M z^{-M}}{M} \sum_{k=0}^{M-1} \frac{H(k)}{1 - rW_M^{-k} z^{-k}}; \quad H(k) = H\left(re^{j2\pi k/M}\right) \tag{6.15}$$

Now approximating $H\left(re^{j2\pi k/M}\right) \approx H\left(e^{j2\pi k/M}\right)$ for $r \approx 1$, we can obtain a stable structure similar to the one in Figure 6.16 containing real values. This is explored in Problem 6.11.

□ **EXAMPLE 6.6** Let $h(n) = \frac{1}{9}\{\underset{\uparrow}{1}, 2, 3, 2, 1\}$. Determine and draw the frequency sampling form.

Solution

MATLAB Script

```
>> h = [1,2,3,2,1]/9;
>> [C,B,A] = dir2fs(h)
C =
    0.5818
    0.0849
    1.0000
B =
   -0.8090    0.8090
    0.3090   -0.3090
A =
    1.0000   -0.6180    1.0000
    1.0000    1.6180    1.0000
    1.0000   -1.0000         0
```

Since $M = 5$ is odd, there is only one first-order section. Hence

$$H(z) = \frac{1 - z^{-5}}{5} \left[0.5818 \frac{-0.809 + 0.809z^{-1}}{1 - 0.618z^{-1} + z^{-2}} + 0.0848 \frac{0.309 - 0.309z^{-1}}{1 + 1.618z^{-1} + z^{-2}} + \frac{1}{1 - z^{-1}} \right]$$

The frequency sampling form is shown in Figure 6.17. □

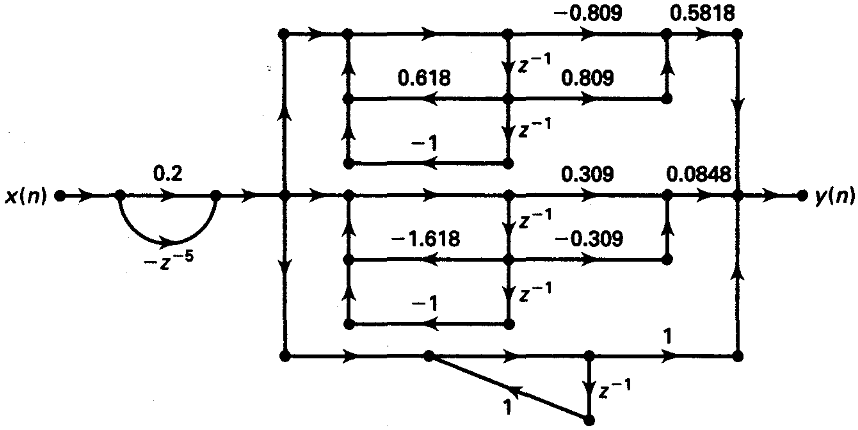

FIGURE 6.17 *Frequency sampling structure in Example 6.6*

□ **EXAMPLE 6.7** The frequency samples of a 32-point linear-phase FIR filter are given by

$$|H(k)| = \begin{cases} 1, & k = 0,1,2 \\ 0.5, & k = 3 \\ 0, & k = 4,5,\ldots,15 \end{cases}$$

Determine its frequency sampling form, and compare its computational complexity with the linear-phase form.

Solution

In this example since the samples of the DFT $H(k)$ are given, we could use (6.13) and (6.14) directly to determine the structure. However, we will use the `dir2fs` function for which we will have to determine the impulse response $h(n)$. Using the symmetry property and the linear-phase constraint, we assemble the DFT $H(k)$ as

$$H(k) = |H(k)|\, e^{j\angle H(k)}, \quad k = 0,1,\ldots,31$$

$$|H(k)| = |H(32-k)|, \quad k = 1,2,\ldots,31;\ H(0) = 1$$

$$\angle H(k) = -\frac{31}{2}\frac{2\pi}{32}k = -\angle H(32-k), \quad k = 0,1,\ldots,31$$

Now the IDFT of $H(k)$ will result in the desired impulse response.

```
>> M = 32; alpha = (M-1)/2;
>> magHk = [1,1,1,0.5,zeros(1,25),0.5,1,1];
>> k1 = 0:15; k2 = 16:M-1;
>> angHk = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
>> H = magHk.*exp(j*angHk);
>> h = real(ifft(H,M));
>> [C,B,A] = dir2fs(h)
```

```
C =
    2.0000
    2.0000
    1.0000
    0.0000
    0.0000
    0.0000
    0.0000
         0
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    1.0000
         0
B =
   -0.9952    0.9952
    0.9808   -0.9808
   -0.9569    0.9569
   -0.8944    0.3162
    0.9794   -0.7121
    0.8265    0.2038
   -0.6754    0.8551
    1.0000    0.0000
    0.6866   -0.5792
    0.5191    0.9883
   -0.4430    0.4993
   -0.8944   -0.3162
   -0.2766    0.3039
    0.9343    0.9996
   -0.9077   -0.8084
A =
    1.0000   -1.9616    1.0000
    1.0000   -1.8478    1.0000
    1.0000   -1.6629    1.0000
    1.0000   -1.4142    1.0000
    1.0000   -1.1111    1.0000
    1.0000   -0.7654    1.0000
    1.0000   -0.3902    1.0000
    1.0000    0.0000    1.0000
    1.0000    0.3902    1.0000
    1.0000    0.7654    1.0000
    1.0000    1.1111    1.0000
    1.0000    1.4142    1.0000
    1.0000    1.6629    1.0000
```

1.0000	1.8478	1.0000
1.0000	1.9616	1.0000
1.0000	-1.0000	0
1.0000	1.0000	0

Note that only four gain coefficients are nonzero. Hence the frequency sampling form is

$$H(z) = \frac{1 - z^{-32}}{32}\left[\begin{array}{c} 2\dfrac{-0.9952 + 0.9952z^{-1}}{1 - 1.9616z^{-1} + z^{-2}} + 2\dfrac{0.9808 - 0.9808z^{-1}}{1 - 1.8478z^{-1} + z^{-2}} + \\ \dfrac{-0.9569 + 0.9569z^{-1}}{1 - 1.6629z^{-1} + z^{-2}} + \dfrac{1}{1 - z^{-1}} \end{array}\right]$$

To determine the computational complexity, note that since $H(0) = 1$, the first-order section requires no multiplication, while the three second-order sections require three multiplications each for a total of nine multiplications per output sample. The total number of additions is 13. To implement the linear-phase structure would require 16 multiplications and 31 additions per output sample. Therefore the frequency sampling structure of this FIR filter is more efficient than the linear-phase structure. □