PENGOLAHAN SINYAL DIGITAL

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

FIR FILTER STRUCTURES

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

FIR FILTER STRUCTURES

- A finite-duration impulse response filter has a system function of the form

$$H(z) = b_o + b_1 z^{-1} + \ldots + b_{M-1} z^{1-M} = \sum_{n=0}^{M-1} b_n z^{-n}$$

- The impulse response *h*(*n*) is

$$h(n) = \begin{cases} b_n, & 0 \le n \le M-1 \\ 0, & \text{else} \end{cases}$$

- The difference equation representation is

$$y(n) = b_o x(n) + b_1 x(n-1) + \ldots + b_{M-1} x(n-M+1)$$

- The order of the filter is *M* – 1, and the *length* of the filter (which is equal to the number of coefficients) is *M*

FIR FILTER STRUCTURES

- The FIR filter structures are
  - a linear convolution of finite support
  - always stable
  - relatively simple compared to IIR structures
- FIR filters can be designed to have a linear-phase response, which is desirable in some applications.

FIR FILTER STRUCTURES

1. **Direct form:** $y(n) = b_o x(n) + b_1 x(n-1) + \ldots + b_{M-1} x(n-M+1)$

2. **Cascade form:** $H(z) = b_o + b_1 z^{-1} + \ldots + b_{M-1} z^{1-M} = \sum_{n=0}^{M-1} b_n z^{-n}$

   this form the system function *H*(*z*) is factored into 2nd-order factors, which are then implemented in a cascade connection.

FIR FILTER STRUCTURES

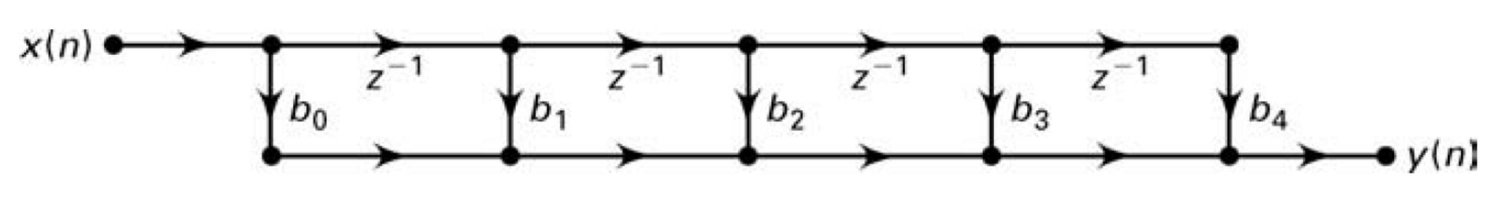
3. **Linear-phase form:**
   its impulse response exhibits certain symmetry conditions
   to reduce multiplications by about half.

4. **Frequency-sampling form:**
   This structure is based on the DFT of the impulse response $h(n)$ and leads to a parallel structure
   suitable for a design technique based on the sampling of frequency response $H(e^{j\omega})$.

DIRECT FORM

- The difference equation is implemented as a tapped delay line since there are no feedback paths.
- Let M = 5 (i.e., a 4th-order FIR filter)

$$y(n) = b_o x(n) + b_1 x(n-1) + b_2 x(n-2) + b_3 x(n-3) + b_4 x(n-4)$$



Direct Form

MATLAB Implementation

- The direct form FIR structure is described by the row vector b containing the *{bn}* coefficients.
- The structure is implemented by the filter function, in which the vector a is set to the scalar value 1

CASCADE FORM

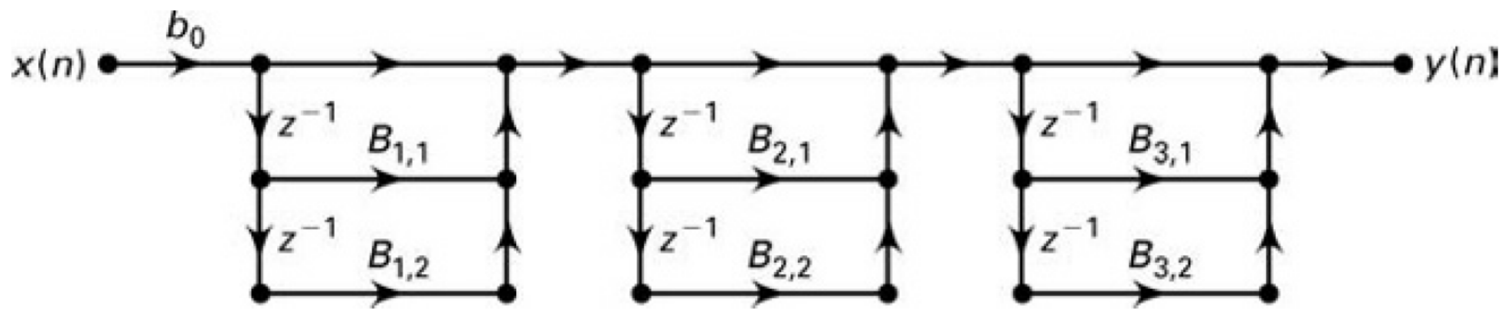
- The system function $H(z)$ is converted into products of 2nd-order sections with real coefficients

$$
\begin{aligned}
H(z) &= b_o + b_1 z^{-1} + \ldots + b_{M-1} z^{-M+1} \\
&= b_o \left(1 + \frac{b_1}{b_o} z^{-1} + \ldots + \frac{b_{M-1}}{b_o} z^{-M+1} \right) \\
&= b_o \prod_{k=1}^{K} \left(1 + B_{k,1} z^{-1} + B_{k,2} z^{-2} \right)
\end{aligned}
$$

- K is equal to $M/2$, and $B_{k,1}$ and $B_{k,2}$ are real numbers representing the coefficients of 2nd-order sections.

CASCADE FORM

- For M = 7 the cascade form is shown as



CASCADE FORM

MATLAB IMPLEMENTATION

- Use our dir2cas function by setting the denominator vector a equal to 1
- Use cas2dir to obtain the direct form from the cascade form.

LINEAR-PHASE FORM

- For frequency-selective filters (e.g., lowpass filters) it is generally desirable to have a phase response that is a linear function of frequency

- where β = 0 or $\pm\pi/2$ and α is a constant.

$$\angle H\left(e^{j\omega}\right) = \beta - \alpha\omega, \quad -\pi < \omega \leq \pi$$

LINEAR-PHASE FORM

- The linear-phase condition imposes the following symmetry conditions on the impulse response *h*(*n*)
  - a symmetric impulse response

$$h(n) = h(M-1-n); \quad \beta = 0, \alpha = \frac{M-1}{2}, \quad 0 \le n \le M-1$$

  - an antisymmetric impulse response

$$h(n) = -h(M-1-n); \quad \beta = \pm\pi/2, \alpha = \frac{M-1}{2}, \quad 0 \le n \le M-1$$

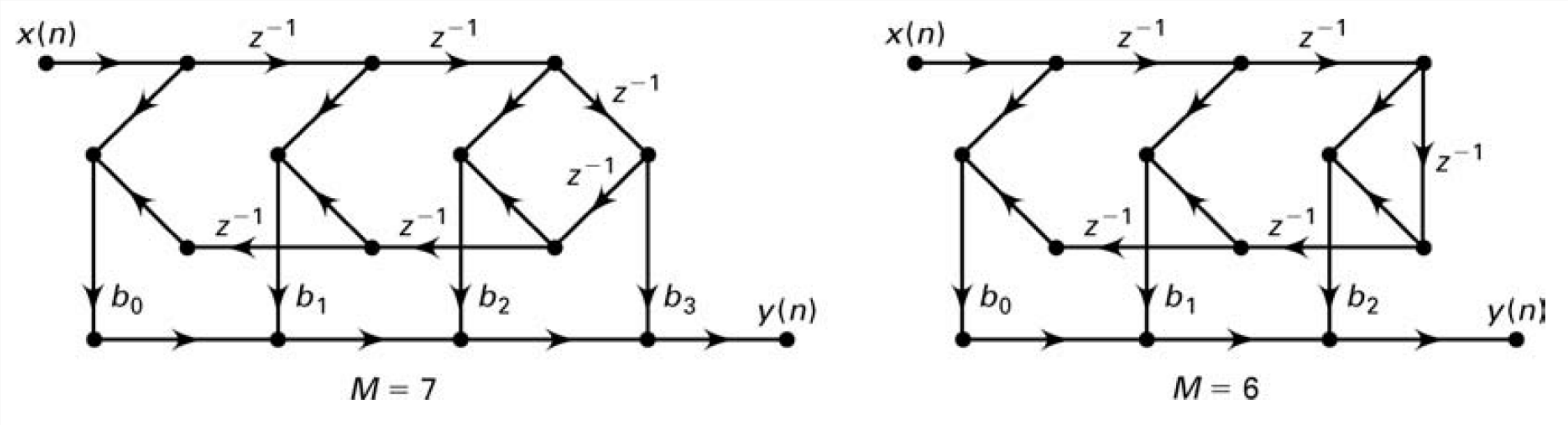
LINEAR-PHASE FORM

- Consider the difference equation with a symmetric impulse response in a symmetric impulse response

$$
\begin{aligned}
y(n) &= b_o x(n) + b_1 x(n-1) + \ldots + b_1 x(n-M+2) + b_o x(n-M+1) \\
&= b_o\left[x(n) + x(n-M+1)\right] + b_1\left[x(n-1) + x(n-M+2)\right] + \ldots
\end{aligned}
$$

LINEAR-PHASE FORM

- The block diagram implementation of the previous difference equation for both odd and even *M*.



LINEAR-PHASE FORM

MATLAB IMPLEMENTATION

- The linear-phase structure is essentially a direct form drawn differently to save on multiplications.
- Hence in a MATLAB representation of the linear-phase structure is equivalent to the direct form.

EXAMPLE

- An FIR filter is given by the system function

$$H(z) = 1 + 16\frac{1}{16}z^{-4} + z^{-8}$$

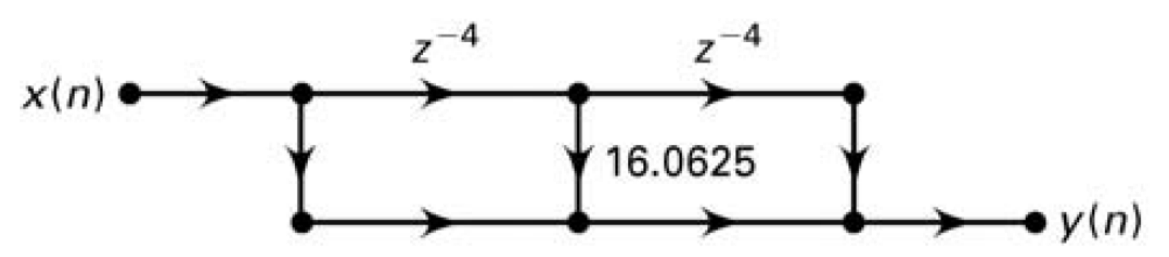
- Determine and draw the direct, linear-phase, and cascade form structures

EXAMPLE

$$H(z) = 1 + 16\frac{1}{16}z^{-4} + z^{-8}$$

- **Direct form:** The difference equation is given by

$$y(n) = x(n) + 16.0625x(n-4) + x(n-8)$$

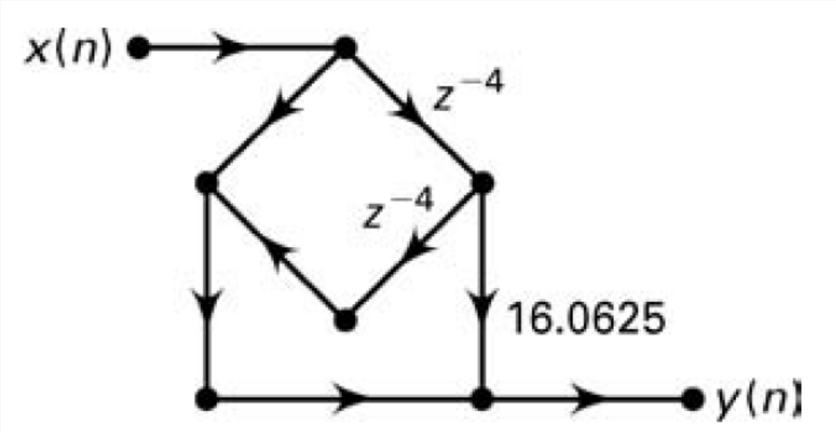


EXAMPLE

$$H(z) = 1 + 16\frac{1}{16}z^{-4} + z^{-8}$$

- **Linear-phase form:** The difference equation can be written in the form

$$y(n) = [x(n) + x(n - 8)] + 16.0625x(n - 4)$$

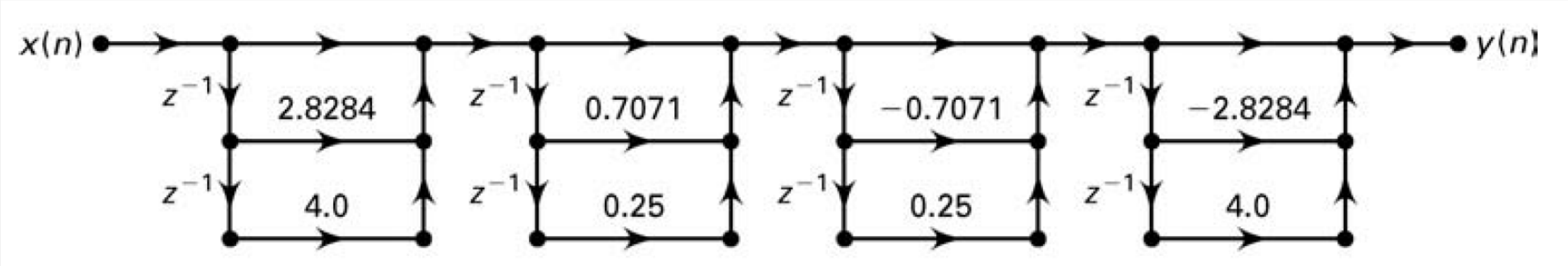


EXAMPLE

$$H(z) = 1 + 16\frac{1}{16}z^{-4} + z^{-8}$$

- **Cascade form:**

>> b=[1,0,0,0,16+1/16,0,0,0,1]; [b0,B,A] = dir2cas(b,1)



LATTICE FILTER STRUCTURES

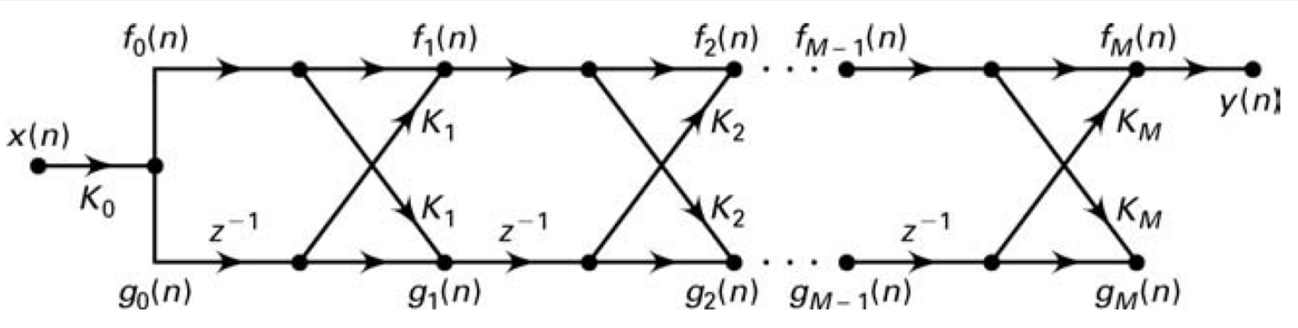
Adhi Harmoko Saputro

LATTICE FILTER STRUCTURES

- Extensively used in digital speech processing and in the implementation of adaptive filters
- The all-zero lattice is the FIR filter representation of the lattice filter, while the lattice ladder is the IIR filter representation.

ALL-ZERO LATTICE FILTERS

- An FIR filter of length M (or order $M - 1$) has a lattice structure with $M-1$ stages



- Each stage of the filter has an input and output that are related by the order-recursive equations

$$f_m(n) = f_{m-1}(n) + K_m g_{m-1}(n-1), \quad m = 1, 2, ..., M-1$$

$$g_m(n) = K_m f_{m-1}(n) + g_{m-1}(n-1), \quad m = 1, 2, ..., M-1$$

- The parameters K_m, $m = 1, 2, \ldots, M - 1$, called the *reflection coefficients*, are the lattice filter coefficients

CONVERSION

- The direct form of the FIR filter

$$H(z) = \sum_{m=0}^{M-1} b_m z^{-m} = b_o \left(1 + \sum_{m=1}^{M-1} \frac{b_m}{b_o} z^{-m} \right)$$

- Computing the lattice filter coefficients $\{K_m\}$

$$K_o = b_o$$

$$K_{M-1} = \alpha_{M-1}(M-1)$$

$$J_m(z) = z^{-m} A_m(z^{-1}), \quad m = M-1, \dots, 1$$

$$A_{m-1}(z) = \frac{A_m(z) - K_m J_m(z)}{1 - K_m^2}, \quad m = M-1, \dots, 1$$

$$K_m = \alpha_m(m), \quad m = M-2, \dots, 1$$

ALL-ZERO LATTICE FILTERS

MATLAB IMPLEMENTATION

```
function [K] = dir2latc(b)
% FIR Direct form to All-Zero Lattice form Conversion
% ---------------------------------------------------
% [K] = dir2latc(b)
% K = Lattice filter coefficients (reflection coefficients)
% b = FIR direct form coefficients (impulse response)
%
M = length(b); K = zeros(1,M); b1 = b(1);
if b1 == 0
      error('b(1) is equal to zero')
end
K(1) = b1; A = b/b1;
for m=M:-1:2
      K(m) = A(m); J = fliplr(A);
      A = (A-K(m)*J)/(1-K(m)*K(m)); A = A(1:m-1);
end
```

ALL-ZERO LATTICE FILTERS

MATLAB IMPLEMENTATION

```
function [y] = latcfilt(K,x)
% LATTICE form realization of FIR filters
% ----------------------------------------
% y = latcfilt(K,x)
% y = output sequence
% K = LATTICE filter (reflection) coefficient array
% x = input sequence
%
Nx = length(x)-1; x = K(1)*x;
M = length(K)-1; K = K(2:M+1); fg = [x; [0 x(1:Nx)]];
for m = 1:M
      fg = [1,K(m);K(m),1]*fg;
      fg(2,:) = [0 fg(2,1:Nx)];
end
y = fg(1,:);
```

ALL-ZERO LATTICE FILTERS

MATLAB IMPLEMENTATION

```
function [b] = latc2dir(K)
% All-Zero Lattice form to FIR Direct form Conversion
% -----------------------------------------------
% [b] = latc2dir(K)
% b = FIR direct form coefficients (impulse response)
% K = Lattice filter coefficients (reflection coefficients)
%
M = length(K); J = 1; A = 1;
for m=2:1:M
A = [A,0]+conv([0,K(m)],J); J = fliplr(A);
end
b=A*K(1);
```

EXAMPLE

- An FIR filter is given by the difference equation

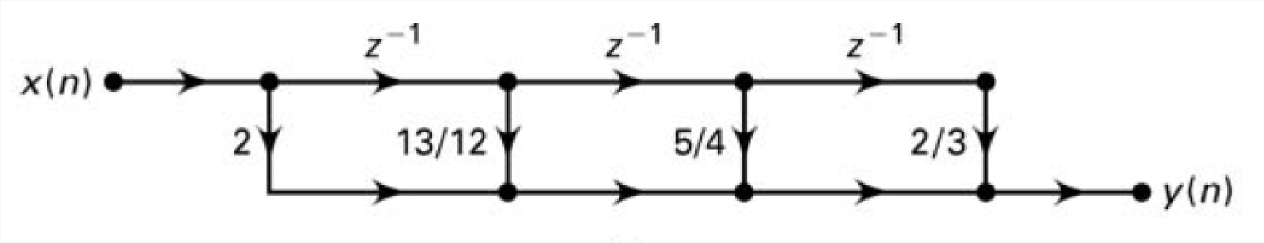
$$y(n) = 2x(n) + \frac{13}{12}x(n-1) + \frac{5}{4}x(n-2) + \frac{2}{3}x(n-3)$$

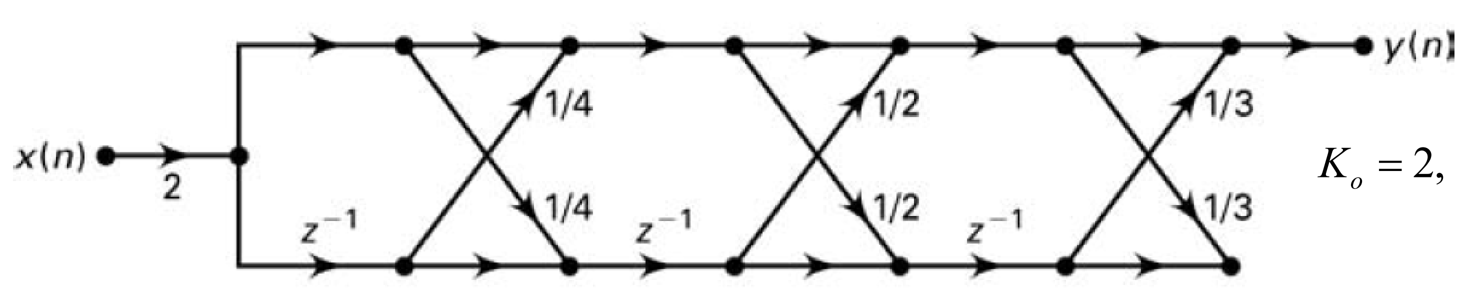
- Determine its lattice form

EXAMPLE

$$y(n) = 2x(n) + \frac{13}{12}x(n-1) + \frac{5}{4}x(n-2) + \frac{2}{3}x(n-3)$$

```
>> b=[2, 13/12, 5/4, 2/3]; K=dir2latc(b)
K =
2.0000 0.2500 0.5000 0.3333
```





$$K_o = 2, \quad K_1 = \frac{1}{4}, \quad K_2 = \frac{1}{2}, \quad K_3 = \frac{1}{3}$$

EXAMPLE

$$y(n) = 2x(n) + \frac{13}{12}x(n-1) + \frac{5}{4}x(n-2) + \frac{2}{3}x(n-3)$$

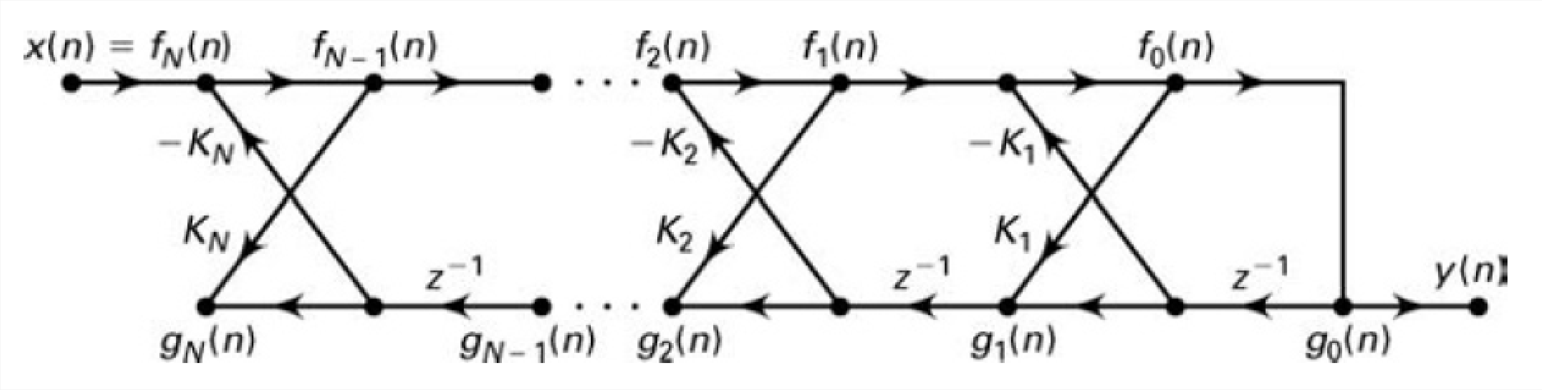
```
>> [x,n] = impseq(0,0,3]; format long; hdirect=filter(b,1,x)
hdirect =
2.00000000000000 1.08333333333333 1.25000000000000 0.66666666666667
>> hlattice=latcfilt(K,x)
hlattice =
2.00000000000000 1.08333333333333 1.25000000000000 0.66666666666667
```

ALL-POLE LATTICE FILTERS

- A lattice structure for an IIR filter is restricted to an all-pole system function
- It can be developed from an FIR lattice structure

$$H(z) = \frac{1}{1 + \sum_{m=1}^{N} a_N(m) z^{-m}}$$

- An *inverse* system to the FIR lattice (*All-pole lattice filter*)



CONVERSION

- The direct form of the FIR filter

$$H(z) = \sum_{m=0}^{M-1} b_m z^{-m} = b_o \left(1 + \sum_{m=1}^{M-1} \frac{b_m}{b_o} z^{-m} \right)$$

- Computing the lattice filter coefficients $\{K_m\}$

$$K_o = b_o$$

$$K_{M-1} = \alpha_{M-1}(M-1)$$

$$J_m(z) = z^{-m} A_m(z^{-1}), \quad m = M-1, \ldots, 1$$

$$A_{m-1}(z) = \frac{A_m(z) - K_m J_m(z)}{1 - K_m^2}, \quad m = M-1, \ldots, 1$$

$$K_m = \alpha_m(m), \quad m = M-2, \ldots, 1$$

ALL-POLE LATTICE FILTER

MATLAB IMPLEMENTATION

- Use the dir2latc function to compute lattice coefficients

- Use the latc2dir function to convert the lattice $\{K_m\}$ coefficients into the direct form $\{a_N(m)\}$

EXAMPLE

- Consider an all-pole IIR filter given by

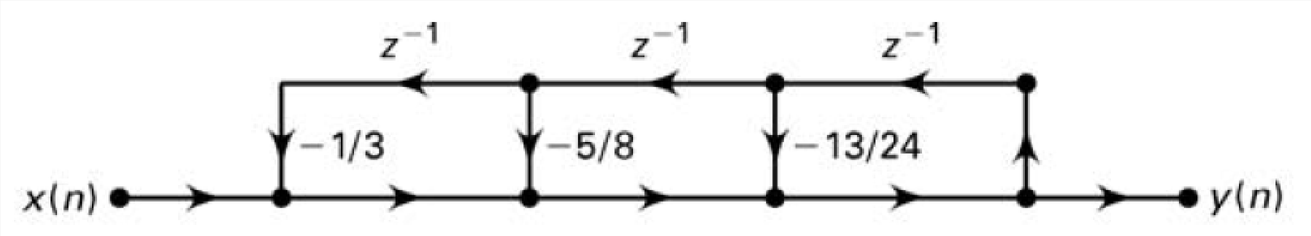
$$H(z) = \frac{1}{1 + \frac{13}{24} z^{-1} + \frac{5}{8} z^{-2} + \frac{1}{3} z^{-3}}$$

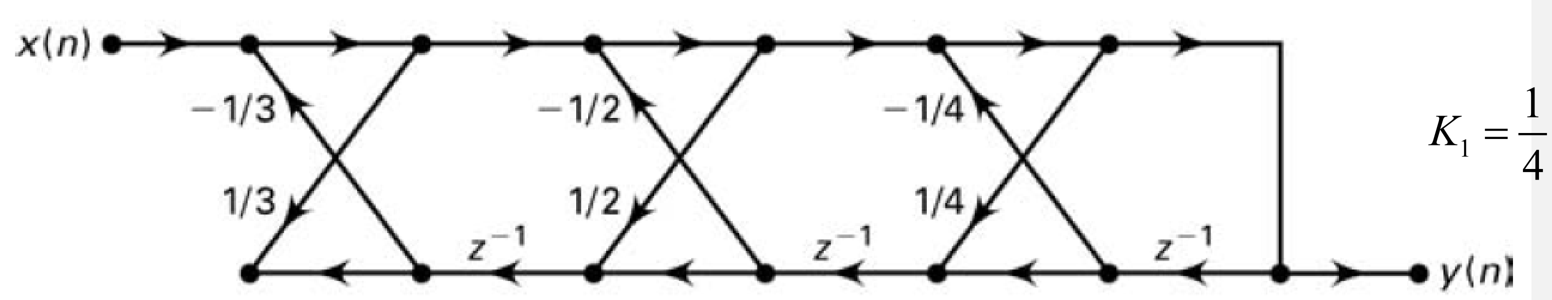
- Determine its lattice structure

EXAMPLE

$$H(z) = \frac{1}{1 + \frac{13}{24} z^{-1} + \frac{5}{8} z^{-2} + \frac{1}{3} z^{-3}}$$

```
>> a=[1, 13/24, 5/8, 1/3]; K=dir2latc(a)
K =
1.0000 0.2500 0.5000 0.3333
```





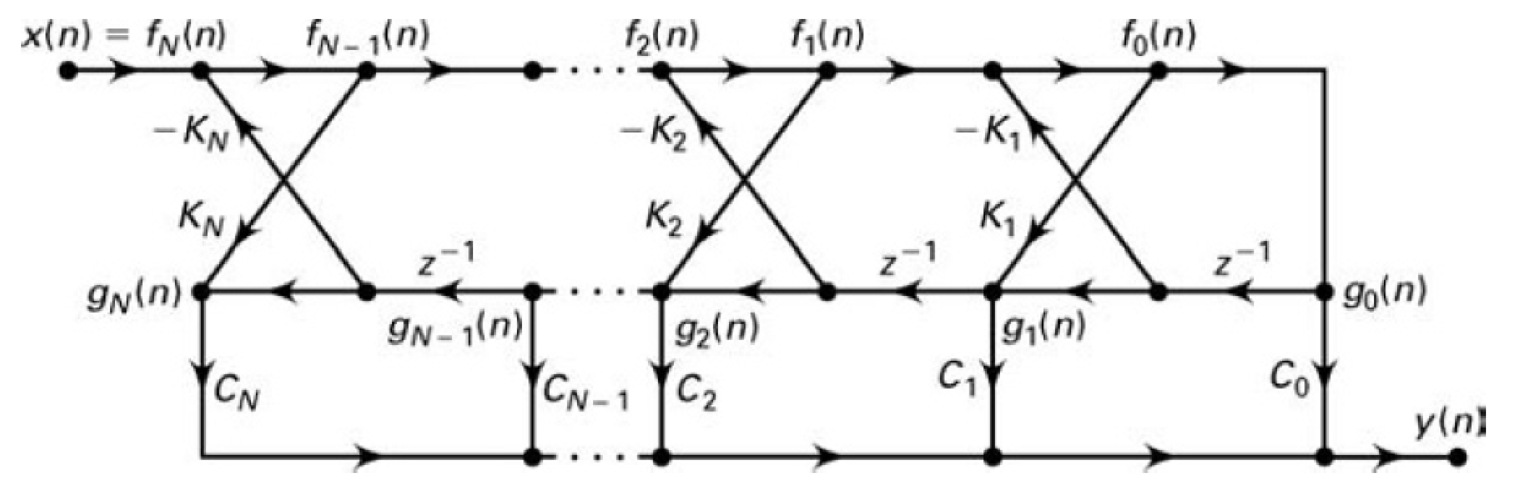
$$K_1 = \frac{1}{4}, \quad K_2 = \frac{1}{2}, \quad K_3 = \frac{1}{3}$$

LATTICE-LADDER FILTERS

- A general IIR filter containing both poles and zeros can be realized as a lattice-type structure by using an all-pole lattice as the basic building block.

$$H(z)=\frac{\sum_{k=0}^{M} b_M(k) z^{-k}}{1+\sum_{k=1}^{N} a_N(k) z^{-k}}=\frac{B_M(z)}{A_N(z)}$$

- The *lattice-ladder* structure



$M = N$

LATTICE-LADDER FILTERS

- The pole-zero IIR filterI output is given by

$$y(n) = \sum_{m=0}^{M} C_m g_m(n)$$

- where $\{C_m\}$ are called the *ladder coefficients* that determine the zeros of the system function $H(z)$.

LATTICE-LADDER FILTERS

MATLAB IMPLEMENTATION

```
function [K,C] = dir2ladr(b,a)
% IIR Direct form to pole-zero Lattice/Ladder form Conversion
% [K,C] = dir2ladr(b,a)
% K = Lattice coefficients (reflection coefficients), [K1,...,KN]
% C = Ladder Coefficients, [C0,...,CN]
% b = Numerator polynomial coefficients (deg <= Num deg)
% a = Denominator polynomial coefficients
a1 = a(1); a = a/a1; b = b/a1; M = length(b); N = length(a);
if M > N
        error(’ *** length of b must be <= length of a ***’)
end
b = [b, zeros(1,N-M)]; K = zeros(1,N-1); A = zeros(N-1,N-1); C = b;
for m = N-1:-1:1
        A(m,1:m) = -a(2:m+1)*C(m+1);
        K(m) = a(m+1); J = fliplr(a);
        a = (a-K(m)*J)/(1-K(m)*K(m)); a = a(1:m);
        C(m) = b(m) + sum(diag(A(m:N-1,1:N-m)));
end
```

LATTICE-LADDER FILTERS

MATLAB IMPLEMENTATION

```
function [b,a] = ladr2dir(K,C)
% Lattice/Ladder form to IIR Direct form Conversion
% [b,a] = ladr2dir(K,C)
% b = numerator polynomial coefficients
% a = denominator polymonial coefficients
% K = Lattice coefficients (reflection coefficients)
% C = Ladder coefficients
N = length(K); M = length(C); C = [C, zeros(1,N-M+1)];
J = 1; a = 1; A = zeros(N,N);
for m=1:1:N
      a = [a,0]+conv([0,K(m)],J); A(m,1:m) = -a(2:m+1); J = fliplr(a);
end
b(N+1) = C(N+1);
for m = N:-1:1
      A(m,1:m) = A(m,1:m)*C(m+1); b(m) = C(m) - sum(diag(A(m:N,1:N-m+1)));
end
```

EXAMPLE

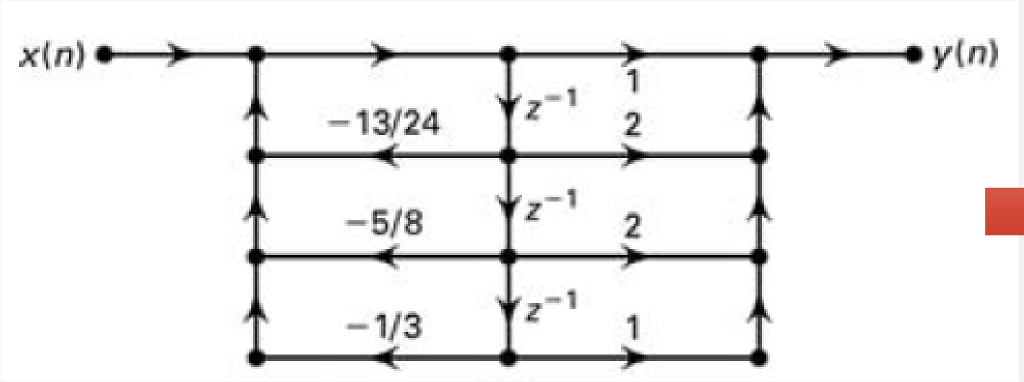
- Convert the following pole-zero IIR filter into a lattice-ladder structure.

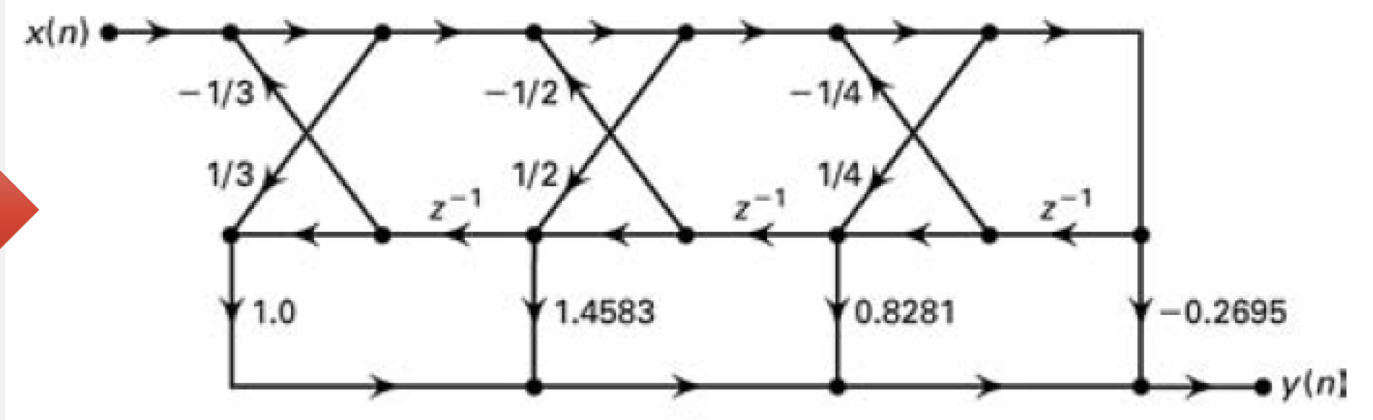
$$H(z)=\frac{1+2z^{-1}+2z^{-2}+z^{-3}}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

EXAMPLE

$$H(z)=\frac{1+2z^{-1}+2z^{-2}+z^{-3}}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

```
>> b = [1,2,2,1] a = [1, 13/24, 5/8, 1/3]; [K,C] = dir2ladrc(b)
K =
0.2500 0.5000 0.3333
C =
-0.2695 0.8281 1.4583 1.0000
```





$$K_1=\frac{1}{4},\quad K_2=\frac{1}{2},\quad K_3=\frac{1}{3}$$

$$C_0=-0.2695,\ C_1=0.8281, C_2=1.4583,\ C_3=1$$

EXAMPLE

$$H(z)=\frac{1+2z^{-1}+2z^{-2}+z^{-3}}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

```
>> [x,n]=impseq(0,0,7) format long hdirect = filter(b,a,x)
hdirect =
Columns 1 through 4
1.00000000000000 1.45833333333333 0.58506944444444 -0.56170428240741
Columns 5 through 8
-0.54752302758488 0.45261700163162 0.28426911049255 -0.25435705167494
>> hladder = ladrfilt(K,C,x)
hladder =
Columns 1 through 4
1.00000000000000 1.45833333333333 0.58506944444444 -0.56170428240741
Columns 5 through 8
-0.54752302758488 0.45261700163162 0.28426911049255 -0.25435705167494
```

TERIMA KASIH

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia