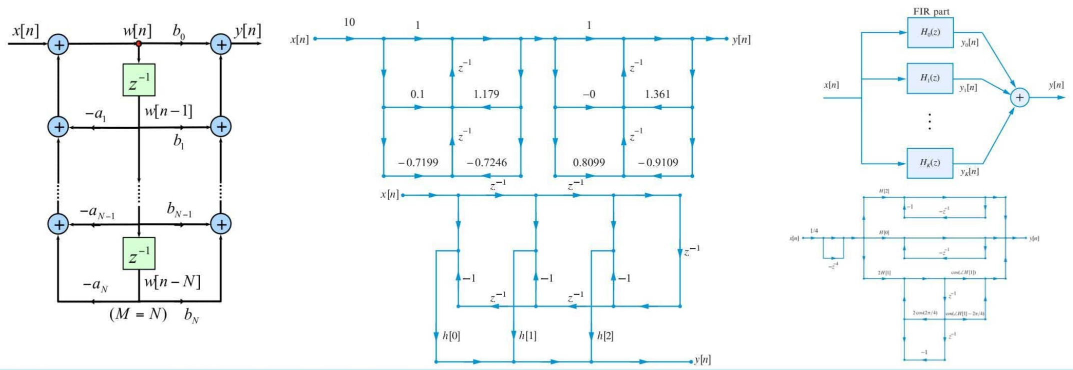
Digital Filter Structures



0

Topics

- Introduction
- Block Diagrams and Signal Flow Graphs (SGF)
  - Motivation for new structures
  - Transposition of SGF
- IIT Filter structures
  - Direct forms I & II
  - Cascade form
  - Parallel form
- FIR Filter Structures
  - Direct form
  - Linear Phase form
  - Cascade form
  - Frequency Sampling form

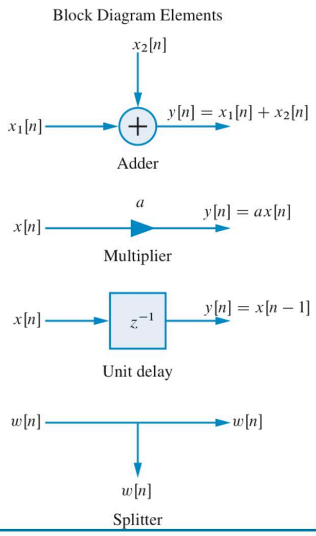
1

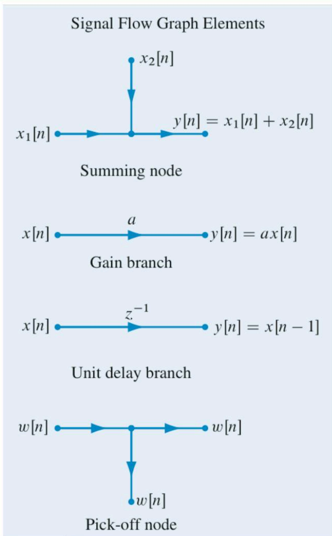
Introduction

- Any practically realizable LTI system can be described by a set of difference equations, which constitute a ***computational algorithm*** for the implementation of the system.
- A ***computational structure*** for a discrete-time system is a pictorial block diagram representation of the computational algorithm using delays, adders, and multipliers.
- A system structure serves as a basis for the development of
  - ***Software*** (that is, a program) that implements the system on a general purpose computer or a special purpose DSP.
  - ***Hardware*** architecture that can be used to implement the system using discrete components or VLSI technology.
- Structures differ in ***computational complexity***, ***memory***, and behavior when we use ***finite precision*** arithmetic.

2

Block Diagrams and Signal Flow Graphs





3

Block Diagrams and Signal Flow Graphs

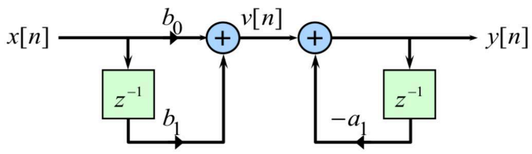
- As an example, consider a first-order discrete-time system

$$y[n] = b_0 x[n] + b_1 x[n-1] - a_1 y[n-1]$$

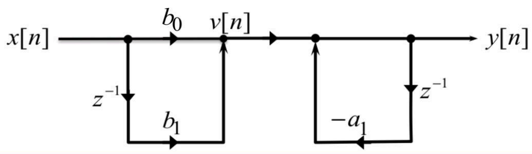
with system function

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}.$$

- Block Diagram:



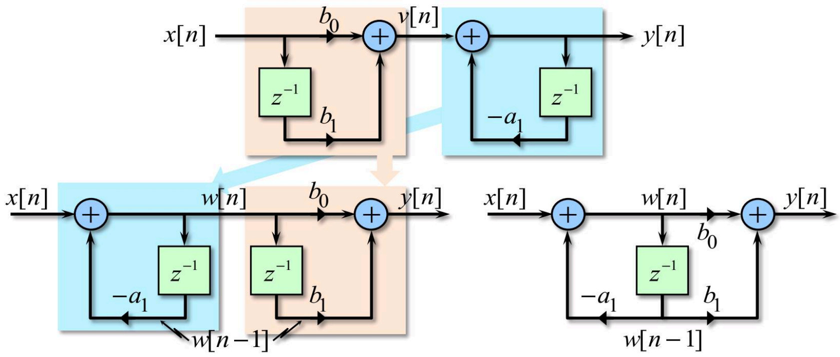
- Signal Flow Graph:



4

Motivation for New Structures

$$y[n] = -a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$

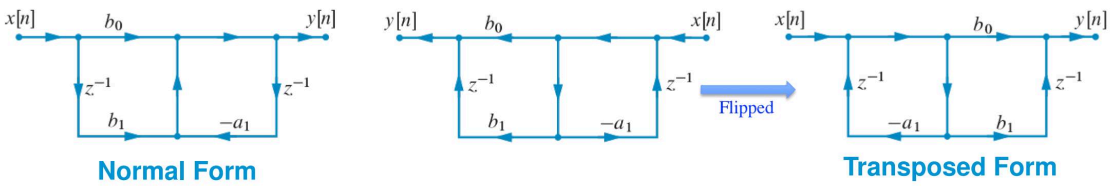


$$w[n] = -a_1 w[n-1] + x[n]$$
$$y[n] = b_0 w[n] + b_1 w[n-1]$$

5

Transposition of Flow Graphs

- This is an alternative structure for the same system function obtained by using a procedure called ***Transposition***:
  - All path arrow directions are reversed.
  - All branch nodes are replaced by adder nodes, and all adder nodes are replaced by branch nodes.
  - The input and output nodes are interchanged.
- The resulting flow graph is called the transposed form flow graph.



IIR Filter Structures

Difference Equation:

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k].$$

System Function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

Three different structures:

- ***Direct Form***: The difference equation is implemented directly as given. Two versions: direct form I and direct form II.
- ***Cascade form***: The system function H(z) is factored into smaller second-order sections (SOS). It is then represented as a product of these SOS. Each SOS is implemented in a direct form II, and the entire system function is implemented as a cascade of SOS.
- ***Parallel form***: After factorization, a partial fraction expansion is used to represent H(z) as a sum of SOS which are implemented in a direct form II, and the system function is implemented as a parallel network of SOS.

IIR Direct Form Structures

Difference Equation: $$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k].$$

System Function: $$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

- **All-Zero:** $$H_1(z) = \sum_{k=0}^{M} b_k z^{-k}$$
- **All-Pole:** $$H_2(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$
- **All-Zero First:** $Y(z) = H_2(z)\,\underbrace{H_1(z)X(z)}_{V(z)} \Rightarrow$ **Direct-Form I**
- **All-Pole First:** $Y(z) = H_1(z)\,\underbrace{H_2(z)X(z)}_{W(z)} \Rightarrow$ **Direct-Form II**

EECE-5666: DSP Digital Filter Structures 8/39

8

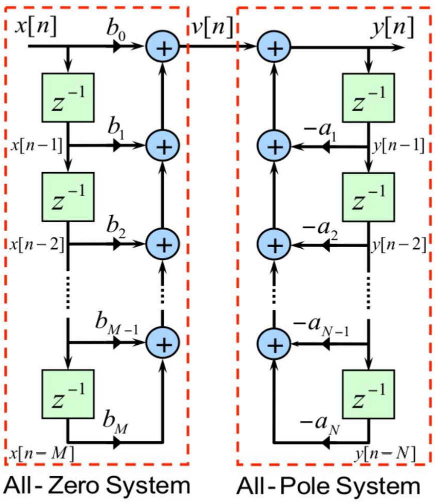
IIR Direct Form I Structure

$$V(z) = \sum_{k=0}^{M} b_k z^{-k} X(z)$$

$$\Rightarrow v[n] = \sum_{k=0}^{M} b_k x[n-k]$$

$$Y(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}} V(z)$$

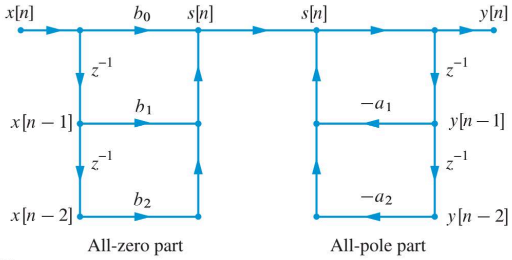
$$\Rightarrow y[n] = -\sum_{k=1}^{N} a_k y[n-k] + v[n]$$



EECE-5666: DSP Digital Filter Structures 9/39

9

Example: IIR Direct Form I



Input-Output Equation

$$y[n] = -a_1 y[n-1] - a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

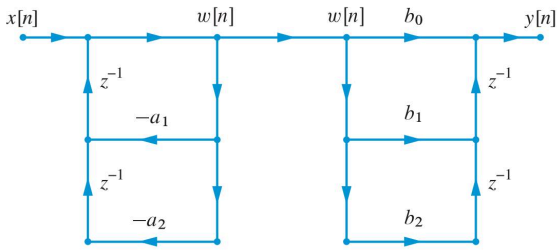
System Function

$$Y(z) = -a_1 z^{-1} Y(z) - a_2 z^{-2} Y(z) + b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z)$$

$$\Rightarrow Y(Z)\left(1 + a_1 z^{-1} + a_2 z^{-2}\right) = X(Z)\left(b_0 + b_1 z^{-1} + b_2 z^{-2}\right)$$

$$\Rightarrow H(Z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Example: Transposed IIR Direct Form I



Input-Output Equations

$$w[n] = -\sum_{k=1}^{N} a_k w[n-k] + x[n],$$

$$y[n] = \sum_{k=0}^{M} b_k w[n-k]$$

Matlab Function filterdf1

```
function [y] = filterdf1(b,a,x)
% Implementation of Direct Form I structure (Normal Form)
%  with zero initial conditions
% [y] = filterdf1(b,a,x)
M = length(b)-1; N = length(a)-1; K = max(M,N);
a0 = a(1); a = reshape(a,1,N+1)/a0;
b = reshape(b,1,M+1)/a0; a = a(2:end);
Lx = length(x); x = [zeros(K,1);x(:)];
Ly = Lx+K; y = zeros(1,Ly);
for n = K+1:Ly
    sn = b*x(n:-1:n-M);
    y(n) = sn - a*y(n-1:-1:n-N);
end
y = y(K+1:Ly);
```


12

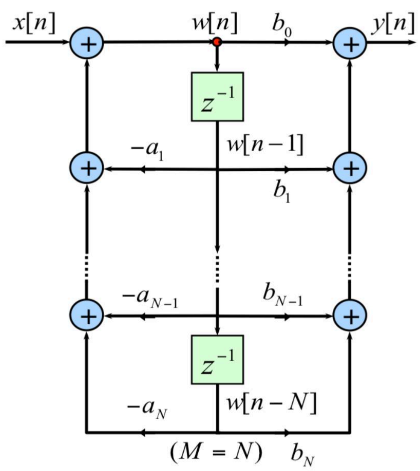
IIR Direct Form II Structure

$$W(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}} X(z)$$

$$\Rightarrow w[n] = -\sum_{k=1}^{N} a_k w[n-k] + x[n]$$

$$Y(z) = \sum_{k=0}^{M} b_k z^{-k} W(z)$$

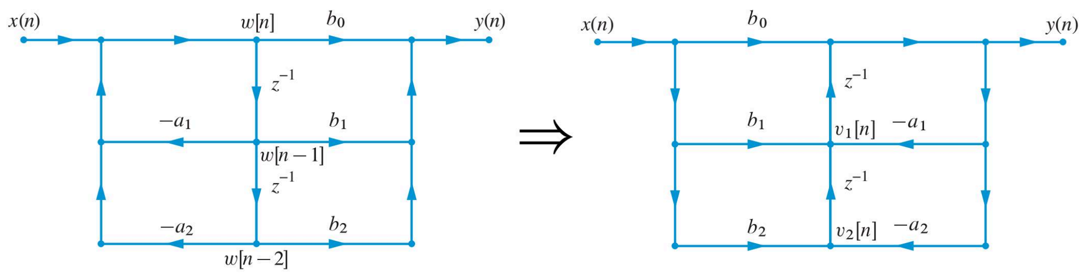
$$\Rightarrow y[n] = \sum_{k=0}^{M} b_k w[n-k]$$



Canonical (minimum memory) Structure

13

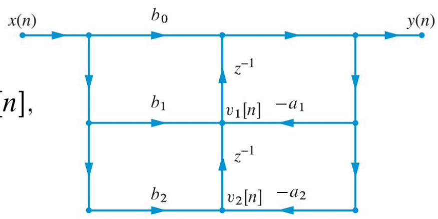
The Transposed Structure



EECE-5666: DSP Digital Filter Structures 14/39

14

Example: Transposed Direct Form II



Input-Output Equation

$$y[n] = v_1[n-1] + b_0 x[n],$$
$$v_1[n] = v_2[n-1] - a_1 y[n] + b_1 x[n],$$
$$v_2[n] = b_2 x[n] - a_2 y[n].$$

System Function

$$Y(z) = z^{-1} V_1(z) + b_0 X(z),$$
$$V_1(z) = z^{-1} V_2(z) - a_1 Y(z) + b_1 X(z),$$
$$V_2(z) = b_2 X(z) - a_2 Y(z)$$

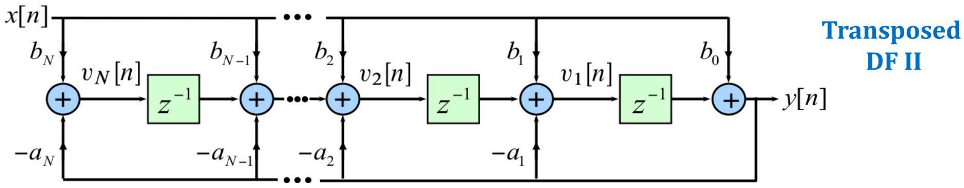
$$\Rightarrow Y(z) = b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) - a_1 z^{-1} Y(z) - a_2 z^{-2} Y(z).$$

$$\Rightarrow \quad H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}},$$

EECE-5666: DSP Digital Filter Structures 15/39

15

MATLAB's `filter` Function



$$y[n] = v_1[n-1] + b_0 x[n]$$
$$v_k[n] = v_{k+1}[n-1] - a_k y[n] + b_k x[n], \quad k = 1\ldots, N-1$$
$$v_N[n] = b_N x[n] - a_N y[n]$$

- `y = filter(b,a,x)`: Implements transposed Direct-form II
- `y = filter(b,a,x,vic)`: Incorporates **Initial Conditions**

$$v_{\mathrm{ic}} = \{v_1[-1], v_2[-1], \ldots, v_N[-1]\}$$

- `vic = filtic(b,a,yic,xic)`: Converts

$$\{y[-1], \ldots, y[-N]; x[-1], \ldots, x[-M]\} \to \{v_1[-1], \ldots, v_{\max(M,N)}[-1]\}$$

EECE-5666: DSP Digital Filter Structures 16/39

16

Matlab Function `filterdf2t`

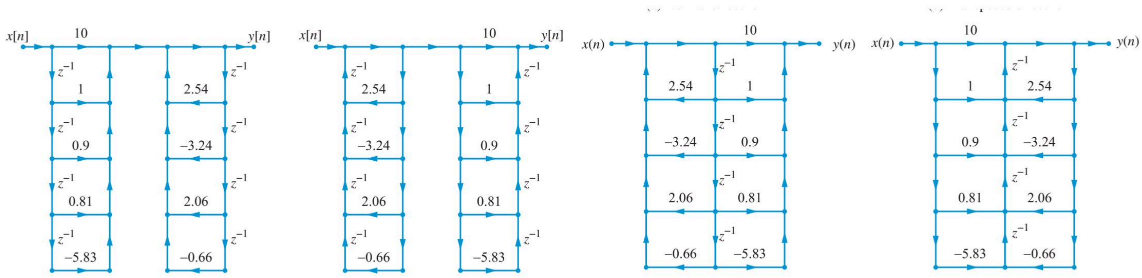
```
function y=filterdf2t(b,a,x,v)
% Implementation of Direct Form II structure (Transposed)
%  with arbitrary initial conditions
% [y] = filterdf2t(b,a,x,v)

N=length(a)-1; M=length(b)-1; K=max(N,M);
L=length(x); y=zeros(L,1);

if nargin < 4, v=zeros(K,1); end
if N>M, b=[b' zeros(1,N-M)]'; else
a=[a' zeros(1,M-N)]'; end

for n=1:L
  y(n)=v(1)+b(1)*x(n);
  for k=1:K-1
    v(k)=v(k+1)-a(k+1)*y(n)+b(k+1)*x(n);
  end
  v(K)=b(K+1)*x(n)-a(K+1)*y(n);
end
```

EECE-5666: DSP Digital Filter Structures 17/39

17

Example: IIR Direct Form Structures

$$H(z) = \frac{10 + z^{-1} + 0.9z^{-2} + 0.8z^{-3} - 5.8z^{-4}}{1 - 2.54z^{-1} + 3.24z^{-2} - 2.06z^{-3} + 0.66z^{-4}}$$



(a) Normal direct form I (b) Transposed direct form I (c) Normal direct form II (d) Transposed direct form II

Cascade Form Structures

- Recall that any rational system function with real coefficients can be expressed in pole-zero form as follows:

$$H(z) = b_0 \frac{\prod_{k=1}^{M_1}\left(1 - z_k z^{-1}\right) \prod_{k=1}^{M_2}\left(1 - z_k z^{-1}\right)\left(1 - z_k^* z^{-1}\right)}{\prod_{k=1}^{N_1}\left(1 - p_k z^{-1}\right) \prod_{k=1}^{N_2}\left(1 - p_k z^{-1}\right)\left(1 - p_k^* z^{-1}\right)},$$

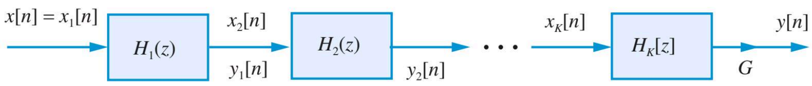
  where $M = M_1 + 2M_2$ and $N = N_1 + 2N_2$.
- After combining any two real real roots and complex-conjugate roots

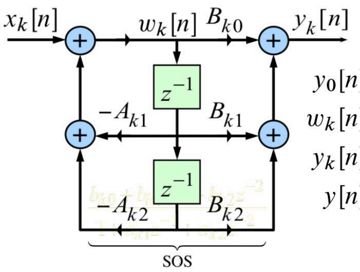
$$H(z) \triangleq G \prod_{k=1}^{K} \underbrace{\frac{B_{k0} + B_{k1}z^{-1} + B_{k2}z^{-2}}{1 + A_{k1}z^{-1} + A_{k2}z^{-2}}}_{\text{Second-Order Section (SOS)}} \triangleq G \prod_{k=1}^{K} H_k(z)$$

  where $b_0 = G \prod_{k=1}^{K} B_{k0}$ and $N = M = 2K$.

Cascade Form Structures

$$H(z) \triangleq G \prod_{k=1}^{K} \underbrace{\frac{B_{k0} + B_{k1} z^{-1} + B_{k2} z^{-2}}{1 + A_{k1} z^{-1} + A_{k2} z^{-2}}}_{\text{Second-Order Section (SOS)}} \triangleq G \prod_{k=1}^{K} H_k(z)$$





Normal Direct-form II Section

$$y_0[n] = x[n]$$
$$w_k[n] = -A_{k1} w_k[n-1] - A_{k2} w_k[n-2] + y_{k-1}[n]$$
$$y_k[n] = B_{k0} w_k[n] + B_{k1} w_k[n-1] + B_{k2} w_k[n-2]$$
$$y[n] = G\, y_K[n] \qquad k = 1, 2, \ldots, K$$

20

Matlab SP Toolbox Functions

MATLAB provides the following functions in its SP Toolbox:

$$\texttt{sos} = \begin{bmatrix} B_{1,0} & B_{11} & B_{12} & 1 & A_{11} & A_{12} \\ B_{2,0} & B_{21} & B_{22} & 1 & A_{21} & A_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ B_{k,0} & B_{K1} & B_{K2} & 1 & A_{K1} & A_{K2} \end{bmatrix}$$

- `[sos,G] = tf2sos(b,a)`

  `G`: Overall gain b_0.

  `sos`: a $K \times 6$ matrix whose kth row contains numerator and denominator coefficients $\{B_{k,\ell}\}$ and $\{A_{k,\ell}\}$; $k = 1, \ldots, K,\ \ell = 0, 1, 2$
- `sos = tf2sos(b,a)` Overall gain b_0 is applied to the first SOS
- `[b,a] = sos2tf(sos,G)`
- `[b,a] = sos2tf(sos)`
  Converts back SOS coefficients to the direct-form coefficients
- `y = sosfilt(sos,x)`
  Simulates the cascade form stricture with input `x` and output `y`

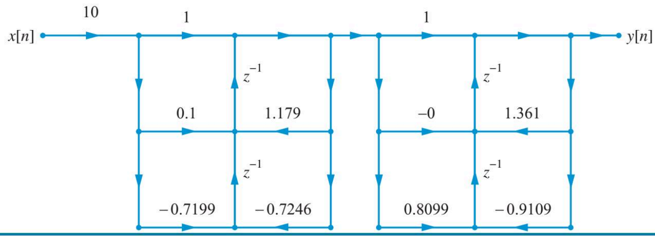
21

Example: Cascade Form Structure

$$H(z) = \frac{10 + z^{-1} + 0.9z^{-2} + 0.8z^{-3} - 5.8z^{-4}}{1 - 2.54z^{-1} + 3.24z^{-2} - 2.06z^{-3} + 0.66z^{-4}}$$

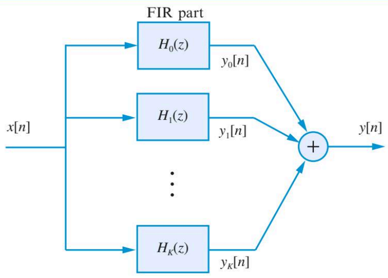
MATLAB script:

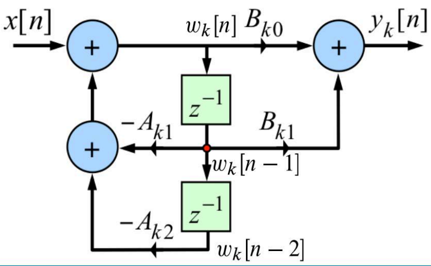
```
>> a = [1 -2.54 3.24 -2.06 0.66];
>> b = [10 1 0.9 0.81 -5.83];
>> [sos,G] = tf2sos(b,a)
sos =
    1.0000    0.1000   -0.7199    1.0000   -1.1786    0.7246
    1.0000   -0.0000    0.8099    1.0000   -1.3614    0.9109
G =
    10
```



22

Parallel Structure





$$H(z) = \underbrace{\sum_{k=0}^{M-N} C_k z^{-k}}_{\text{FIR part}} + \underbrace{\sum_{k=1}^{K} \frac{A_k}{1 - p_k z^{-1}}}_{\text{Distinct roots}}$$

$$\frac{A_k}{1 - p_k z^{-1}} + \frac{A_k^*}{1 - p_k^* z^{-1}} = \frac{B_{k0} + B_{k1} z^{-1}}{1 + A_{k1} z^{-1} + A_{k2} z^{-2}} \triangleq H_k(z)$$

$$w_k[n] = -A_{k1} w_k[n-1] - A_{k2} w[n-2] + x[n],$$
$$y_k[n] = B_{k0} w_k[n] + B_{k1} w_k[n-1], \ k = 1, \ldots, K$$
$$y[n] = \sum_{k=0}^{M-N} C_k x[n-k] + \sum_{k=1}^{K} y_k[n].$$

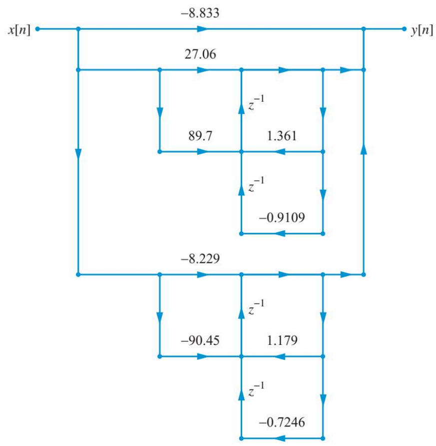
23

Example: Parallel Form Structure

$$H(z) = \frac{10 + z^{-1} + 0.9z^{-2} + 0.8z^{-3} - 5.8z^{-4}}{1 - 2.54z^{-1} + 3.24z^{-2} - 2.06z^{-3} + 0.66z^{-4}}$$

MATLAB script:

```
>> b = [10 1 0.9 0.81 -5.83]; a = [1 -2.54 3.24 -2.06 0.66];
>> [R,p,C] = residuez(b,a); C
C =
   -8.8333
>> [B1,A1] = residuez(R(1:2),p(1:2),[]);
>> B1 = real(B1)
B1 =
   27.0624   89.7028
>> A1 = real(A1)
A1 =
    1.0000   -1.3614    0.9109
>> [B2,A2] = residuez(R(3:4),p(3:4),[]);
>> B2 = real(B2)
B2 =
   -8.2291  -90.4461
>> A2 = real(A2)
A2 =
    1.0000   -1.1786    0.7246
```



MATLAB Functions from I&P Book

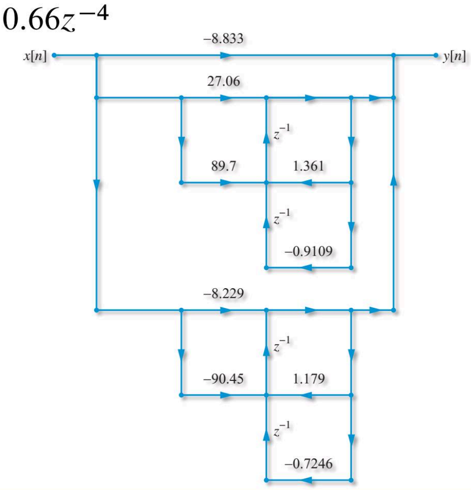
MATLAB does not have functions for parallel form. The following functions are from I&P (DSPUM) book.

- `[C,B,A] = dir2par(b,a)`
  `C`: Polynomial part coefficient
  `B`: $K \times 2$ matrix containing numerator coefficients $\{b_{k,\ell}\}$
  `A`: $K \times 3$ matrix containing numerator coefficients $\{a_{k,\ell}\}$
- `[b,a] = dir2par(C,B,A)` par2dir
  Converts back parallel form to the direct form coefficients.
- `y = parfiltr(C,B,A,x)`
  Simulates the parallel form structure with input `x` and output `y`.

Example: Parallel Form Structure

$$H(z) = \frac{10 + z^{-1} + 0.9z^{-2} + 0.8z^{-3} - 5.8z^{-4}}{1 - 2.54z^{-1} + 3.24z^{-2} - 2.06z^{-3} + 0.66z^{-4}}$$

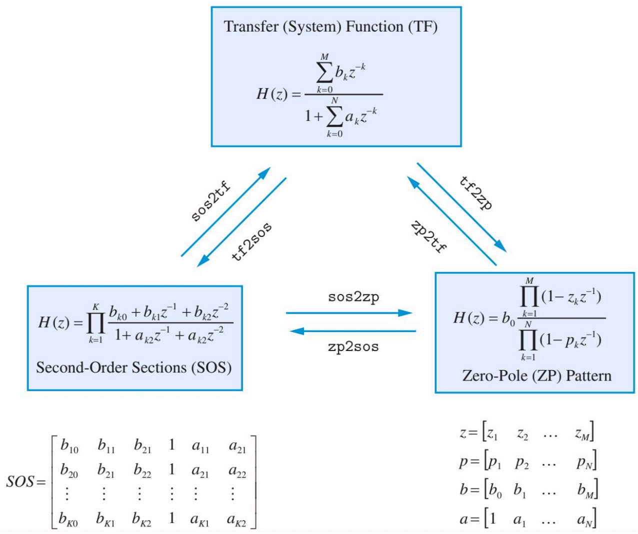
```
>> b = [10,1,0.9,0.8,-5.8];
>> a = [1,-2.54,3.24,-2.06,0.66];
>> [C,B,A] = dir2par(b,a)
C =
    -8.7879
B =
    -8.2622   -90.2143
    27.0501    89.5268
A =
      1.0000   -1.1786     0.7246
      1.0000   -1.3614     0.9109
```



EECE-5666: DSP Digital Filter Structures 26/39

26

Structure Conversions in SP Toolbox



EECE-5666: DSP Digital Filter Structures 27/39

27

FIR Filter Structures

Difference Equation : $y[n] = \sum_{k=0}^{M} b_k x[n-k]$

Impulse Response : $h[n] = \begin{cases} b_n, & n = 0, 1, \ldots, M \\ 0, & \text{otherwise} \end{cases}$

System Function : $H(z) = \dfrac{Y(z)}{X(z)} = \sum_{n=0}^{M} b_n z^{-n}$

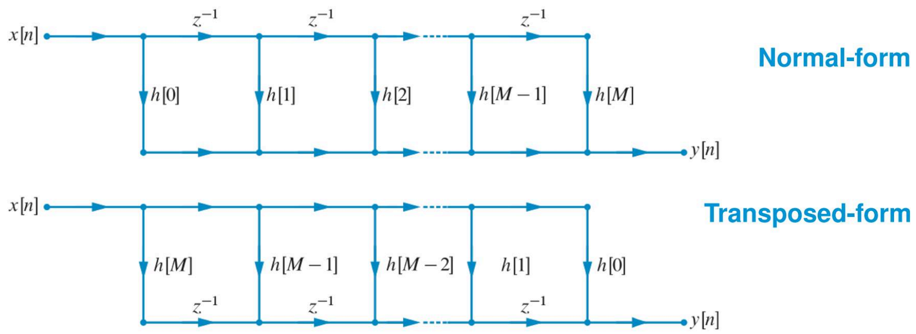
Four different structures:

- **Direct form:** In this form the difference equation is implemented directly as given.
- **Cascade form:** Similar to IIR cascade form.
- **Linear-phase form:** When an FIR filter has a linear-phase response, its impulse response exhibits even/odd symmetries which are exploited to reduce multiplications by about half.
- **Frequency-sampling form:** A form similar to the IIR parallel form using the DFT, useful for narrowband FIR filters.

FIR Direct Form

- Implement the difference equation directly

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \cdots + h[M]x[n-M]$$

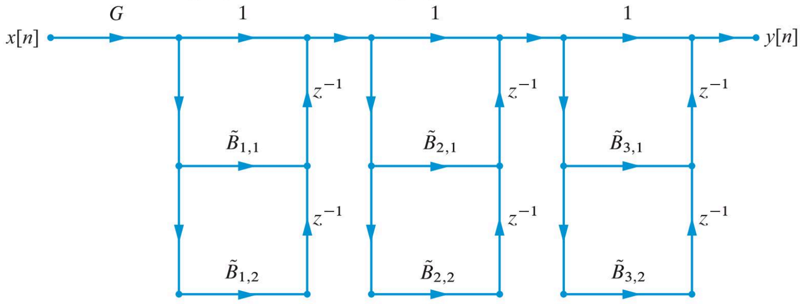


- Also known as a tapped delay-line or a transversal filter.
- Matlab implementation is using the familiar `filter` function with `a = 1`:
  `y = filter(b,1,x)`

FIR Cascade Form

- This form is similar to the IIR form and is obtained using factorization of the system function into second-order sections:

$$H(z) = \sum_{n=0}^{M} h[n]z^{-n} \triangleq G \prod_{k=0}^{K} \left(1 + \tilde{B}_{k1}z^{-1} + \tilde{B}_{k2}z^{-2}\right)$$



- MATLAB Implementation: The previously discussed functions `tf2sos`, `sos2tf`, and `sosfilt` can be used with `a = 1`.

30

Direct Form for Linear-Phase FIR Filters

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

$$\angle H(\omega) = \beta - \alpha\omega, -\pi < \omega \le \pi; \ \beta = 0, \text{ or } \pm\pi/2, \ \alpha \text{ is a constant}$$

- Then the impulse response satisfies:

$$h(n) = h(M-n); \ \beta = 0, \ \alpha = M/2, \ 0 \le n \le M$$

or

$$h(n) = -h(M-n); \ \beta = \pm\pi/2, \ \alpha = M/2, \ 0 \le n \le M$$

- Consider the difference equation for symmetric impulse response

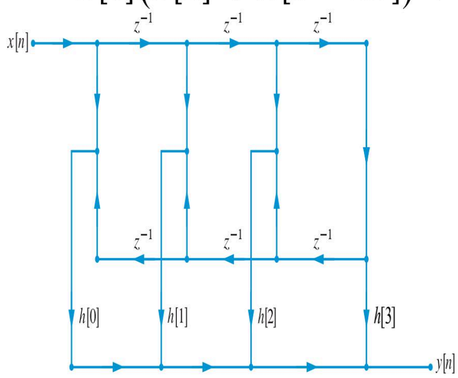
$$\begin{aligned} y(n) &= h(0)x(n) + h(1)x(n-1) + \cdots + h(1)x(n-M+1) + h(0)x(n-m) \\ &= h(0)\big(x(n) + x(n-M)\big) + h(1)\big(x(n-1) + x(n-M+1)\big) + \cdots \end{aligned}$$

which adds corresponding signals first and then performs the required multiplication with the filter coefficients,

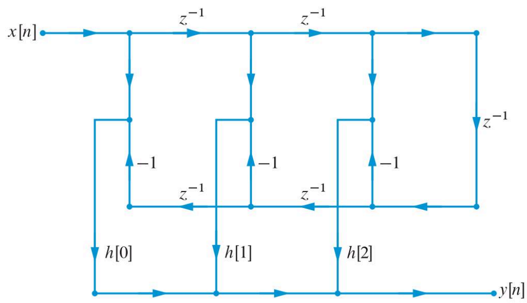
31

Direct Form for Linear-Phase FIR Filters

$$y[n] = h[0]x[n] + h[1]x[n-1] + \cdots + h[1]x[n-M+1] + h[0]x[n-M]$$
$$= h[0]\big(x[n] + x[n-M]\big) + h[1]\big(x[n-1] + x[n-M+1]\big) + \cdots$$



Even $M (= 6)$



Odd $M (= 5)$

32

Example on FIR Filter Structures

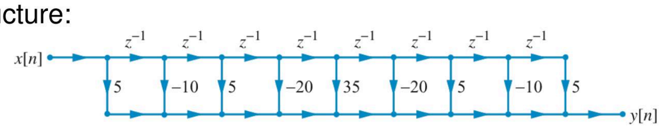
Consider the following system function $H(z)$ of a linear-phase FIR system:

$$H(z) = 5 - 10z^{-1} + 5z^{-2} - 20z^{-3} + 35z^{-4} - 20z^{-5} + 5z^{-6} - 10z^{-7} + 5z^{-8}.$$

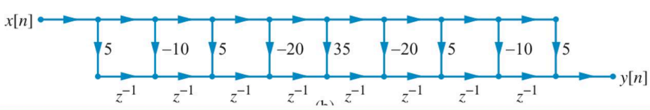
- **Direct-form:** The difference equation is

$$\begin{aligned} y[n] \quad = \quad & 5x[n] - 10x[n-1] + 5x[n-2] - 20x[n-3] + 35x[n-4] \\ & - 20x[n-5] + 5x[n-6] - 10x[n-7] + 5x[n-8]. \end{aligned}$$

  - Normal Structure:



  - Transposed structure:



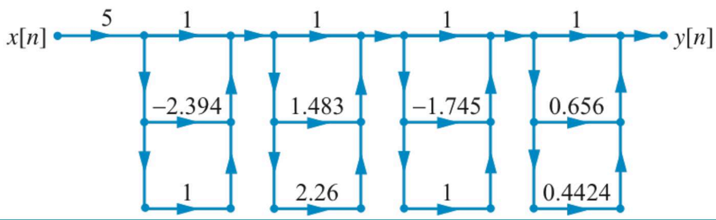
33

Example on FIR Filter Structures

$$H(z) = 5-10z^{-1}+5z^{-2}-20z^{-3}+35z^{-4}-20z^{-5}+5z^{-6}-10z^{-7}+5z^{-8}.$$

- **Cascade-form:** Using the `tf2sos` function

```
>> [sos,G] = tf2sos(b,a)
>> sos =
    1.0000   -2.3940    1.0000    1.0000         0         0
    1.0000    1.4829    2.2604    1.0000         0         0
    1.0000   -1.7450    1.0000    1.0000         0         0
    1.0000    0.6560    0.4424    1.0000         0         0
G =
    5
```



EECE-5666: DSP Digital Filter Structures 34/39

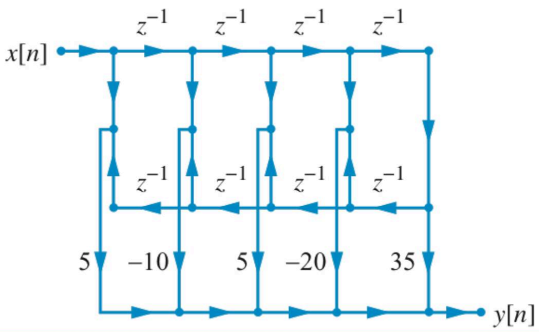
34

Example on FIR Filter Structures

$$H(z) = 5-10z^{-1}+5z^{-2}-20z^{-3}+35z^{-4}-20z^{-5}+5z^{-6}-10z^{-7}+5z^{-8}.$$

- **Linear-Phase form:** The difference equation can be written as

$$\begin{aligned} y[n] \quad &= \quad 5\big(x[n]+x[n-8]\big)-10\big(x[n-1]+x[n-7]\big) \\ &\quad + 5\big(x[n-2]+x[n-6]\big)-20\big(x[n-3]+x[n-5]\big)+35x[n-4]. \end{aligned}$$



EECE-5666: DSP Digital Filter Structures 35/39

35

FIR Frequency Sampling Form

- We use the fact that the system function $H(z)$ of an FIR filter can be reconstructed from its samples, $H[k] = H(z)\big|_{z=e^{j\frac{2\pi k}{N}}}$, on the unit circle.
- In (7.72) we showed that

$$H(z) = \mathcal{Z}\{h[n]\} = \mathcal{Z}\Big[\text{IDFT}\{h[k]\}\Big] = \mathcal{Z}\left[\frac{1}{N}\sum_{k=0}^{N-1} H[k]W_N^{-nk}\right]$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} H[k]\left[\sum_{n=0}^{N-1}\left(z^{-1}W_N^{-k}\right)^n\right] = \frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H[k]}{1-z^{-1}e^{j\frac{2\pi}{N}k}}$$

- One problem with this formulation 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 $e^{j2\pi k/N}$ factor.

FIR Frequency Sampling Form

- By combining complex-conjugate pairs, it can be shown that

$$H(z) = \frac{1-z^{-N}}{N}\left\{\frac{H[0]}{1-z^{-1}} + \frac{H\left[\frac{N}{2}\right]}{1+z^{-1}} + \sum_{k=1}^{K} 2\big|H[k]\big|H_k(z)\right\},$$

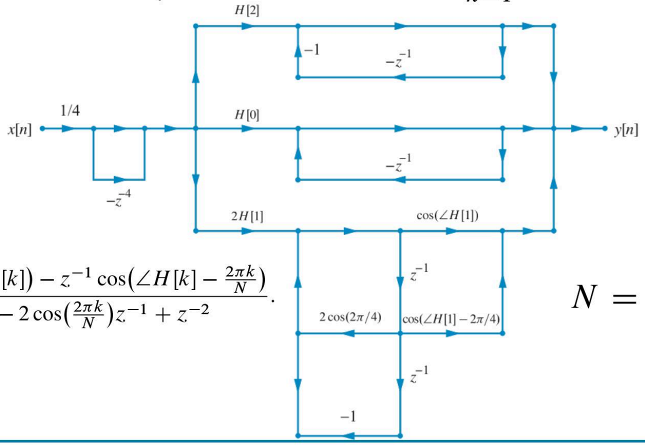
where $K = N/2 - 1$ for N even or $K = (N-1)/2$ for N odd, and where the second-order sections $H_k(z)$, for $k = 1, 2, \ldots, K$ are given by

$$H_k(z) = \frac{\cos\big(\angle H[k]\big) - z^{-1}\cos\big(\angle H[k] - \frac{2\pi k}{N}\big)}{1 - 2\cos\big(\frac{2\pi k}{N}\big)z^{-1} + z^{-2}}.$$

- The derivation is left as an exercise..

FIR Frequency Sampling Form

$$H(z) = \frac{1 - z^{-N}}{N} \left\{ \frac{H[0]}{1 - z^{-1}} + \frac{H\left[\frac{N}{2}\right]}{1 + z^{-1}} + \sum_{k=1}^{K} 2\left|H[k]\right| H_k(z) \right\},$$



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

$$N = 4 = M + 1$$

38

Example on Frequency Sampling FIR Filter Structures

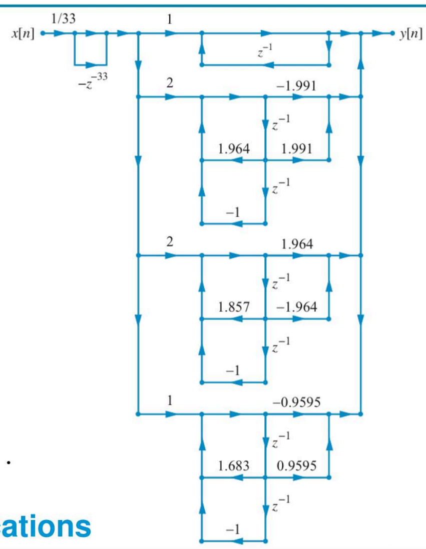
Let

$$H[k] = H\left(e^{j\frac{2\pi}{33}k}\right)$$

$$= e^{-j\frac{32\pi}{33}k} \times \begin{cases} 1, & k = 0, 1, 2, 31, 32 \\ 0.5, & k = 3, 30 \\ 0. & \text{otherwise} \end{cases}$$

Then we obtain

$$\begin{aligned} H(z) &= \frac{1 - z^{-33}}{33}\left[\frac{1}{1 - z^{-1}} + \frac{-1.99 + 1.99z^{-1}}{1 - 1.964z^{-1} + z^{-2}}\right. \\ &\quad + \left.\frac{1.964 - 1.964z^{-1}}{1 - 1.857z^{-1} + z^{-2}} + \frac{-1.96 + 1.96z^{-1}}{1 - 1.683z^{-1} + z^{-2}}\right]. \end{aligned}$$



Requires only 9 multiplications

39