DSP: Part II

Lecture.5.1

DIGITAL FILTER STUCTURE - OVERVIEW

WHY DO WE CONSIDER VARIOUS FILTER STRUCTURES?

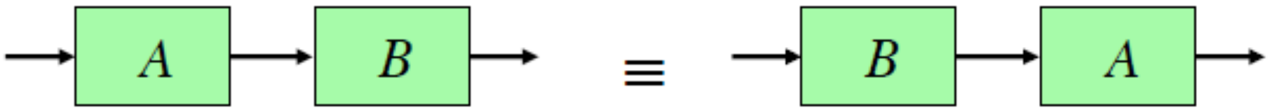
- The consideration of different structure is to:

  - save computation cost, save delay

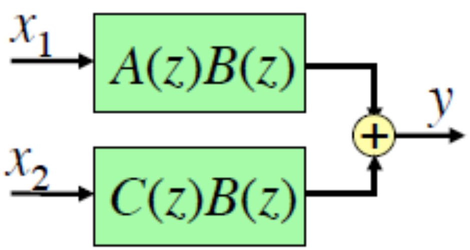
  - more robust to quantization effects

- If system is implemented in double precision, than all filter structure will yield the same result!

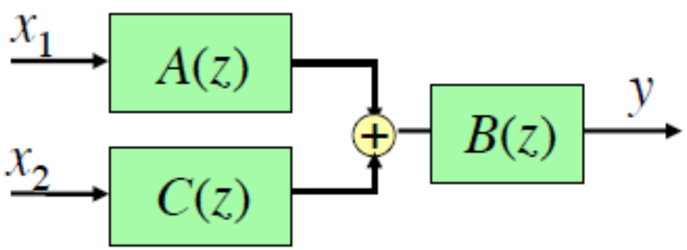
  - e.g. Commutation $H = AB = BA$



  - Factoring $AB + CB = (A + C) \cdot B$



fewer blocks



less computation

1) OVERVIEW: DIGITAL FILTER STRUCTURES

Filter: $H(z) = \frac{B(z)}{A(z)}$ with input $x[n]$ and output $y[n]$

$$y[n] = \sum_{k=0}^{M} b[k]x[n-k] - \sum_{k=1}^{N} a[k]y[n-k]$$

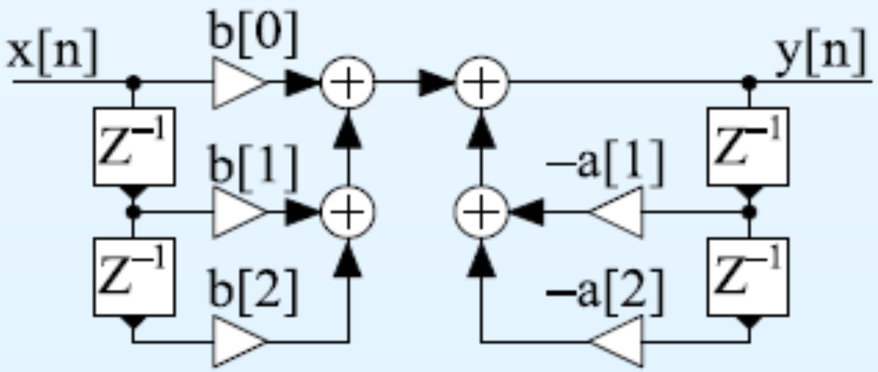
Direct forms use coefficients $a[k]$ and $b[k]$ directly

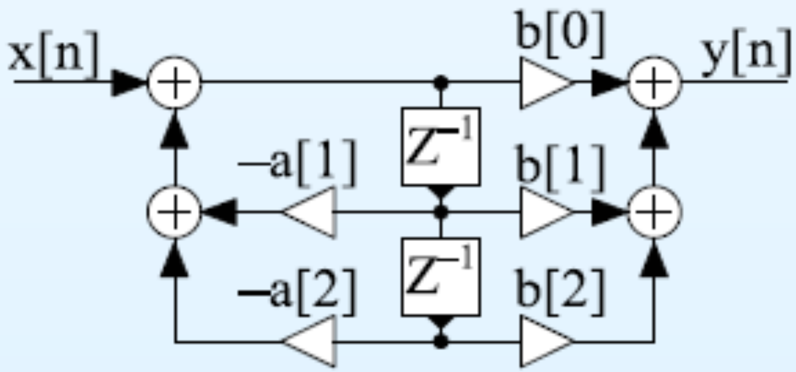
Direct Form 1:

- Direct implementation of difference equation
- Can view as $B(z)$ followed by $A(z)$

Direct Form II:

- Implements $A(z)$ followed by $B(z)$
- Saves on delays (=storage)





Transposition

Can convert any block diagram into an equivalent transposed form:

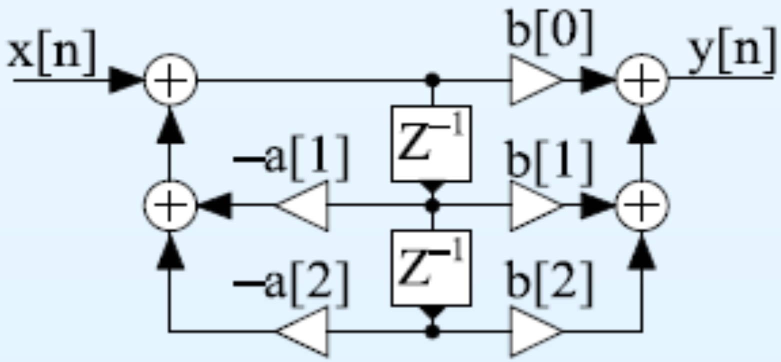
- Reverse direction of each interconnection
- Reverse direction of each multiplier
- Change junctions to adders and vice-versa
- Interchange the input and output signals

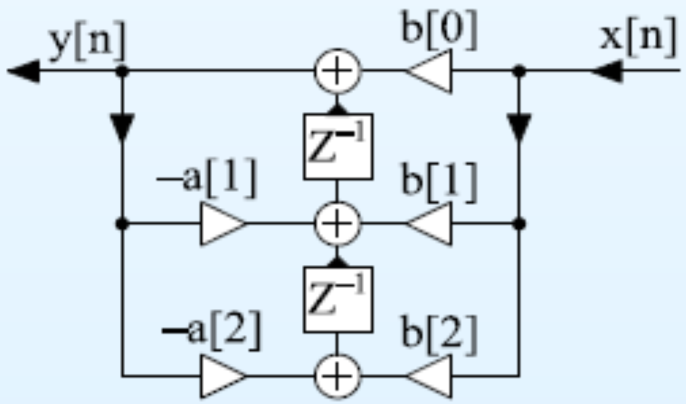
Example:

Direct form II $\rightarrow$ Direct Form II$_t$

Would normally be drawn with input on the left

Note: A valid block diagram must never have any feedback loops that don't go through a delay (z^{-1} block).





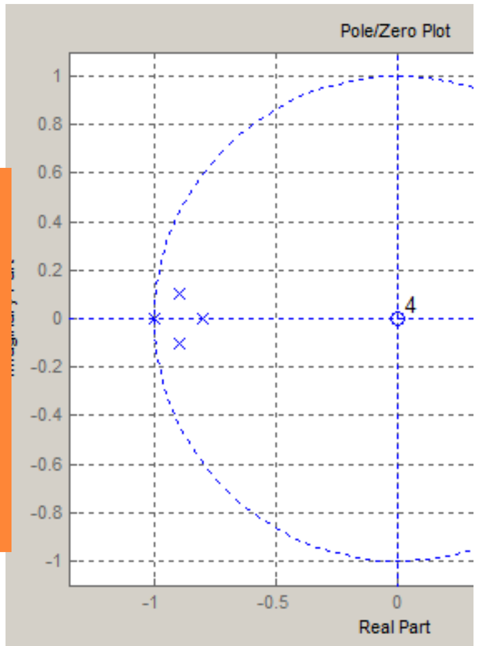
QUANTIZATION ON COEFFICIENTS

- When the coefficients of a filter is quantized, it will shift the location of the poles, zeros. This is severe especially for higher polynomials. E.g, 4^{th} order poles

$$H1(z) = \frac{z^4}{(z+0.9)^4} = \frac{1}{1+3.6z^{-1}+4.86z^{-2}+2.916z^{-3}+\mathbf{0.6561z^{-4}}}$$

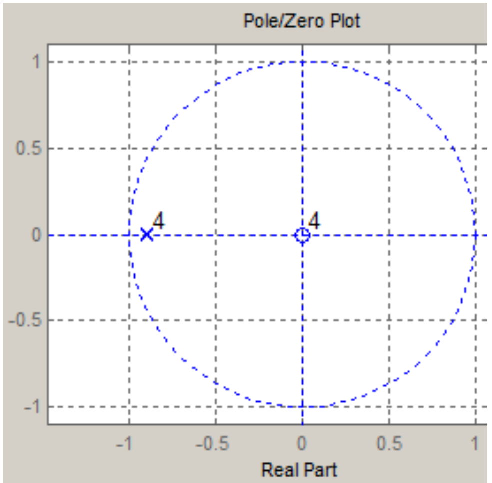
$$H2(z) = \frac{1}{1+3.6z^{-1}+4.86z^{-2}+2.916z^{-3}+\mathbf{0.656z^{-4}}}$$

```
>> roots([1 3.6 4.86 2.916 0.656])

ans =

  -1.0000
  -0.9000 + 0.1000i
  -0.9000 - 0.1000i
  -0.8000
```

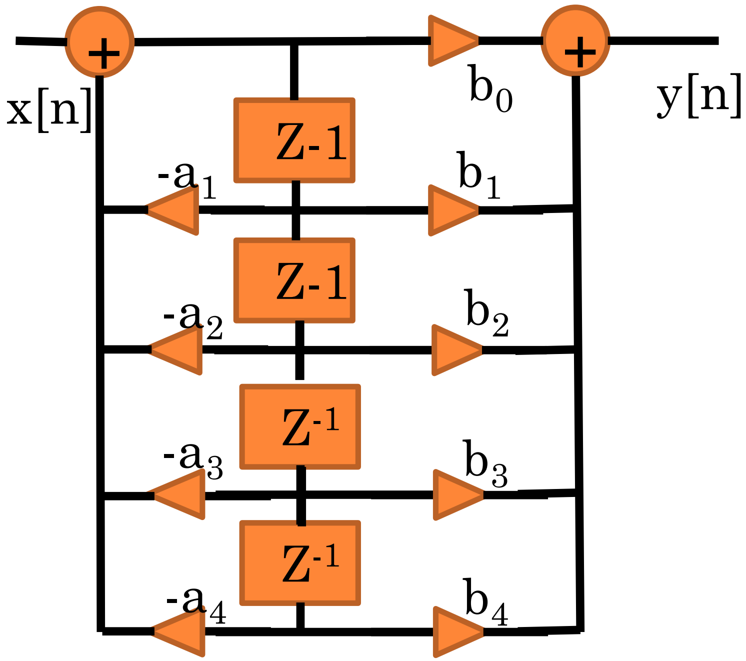


One pole has gone to unit circle just bcos of quantization of 0.6561 to 0.656 at z^{-4}

```
>> roots( [1 3.6 4.86 2.916 0.6561])

ans =

  -0.9002
  -0.9000 + 0.0002i
  -0.9000 - 0.0002i
  -0.8998
```

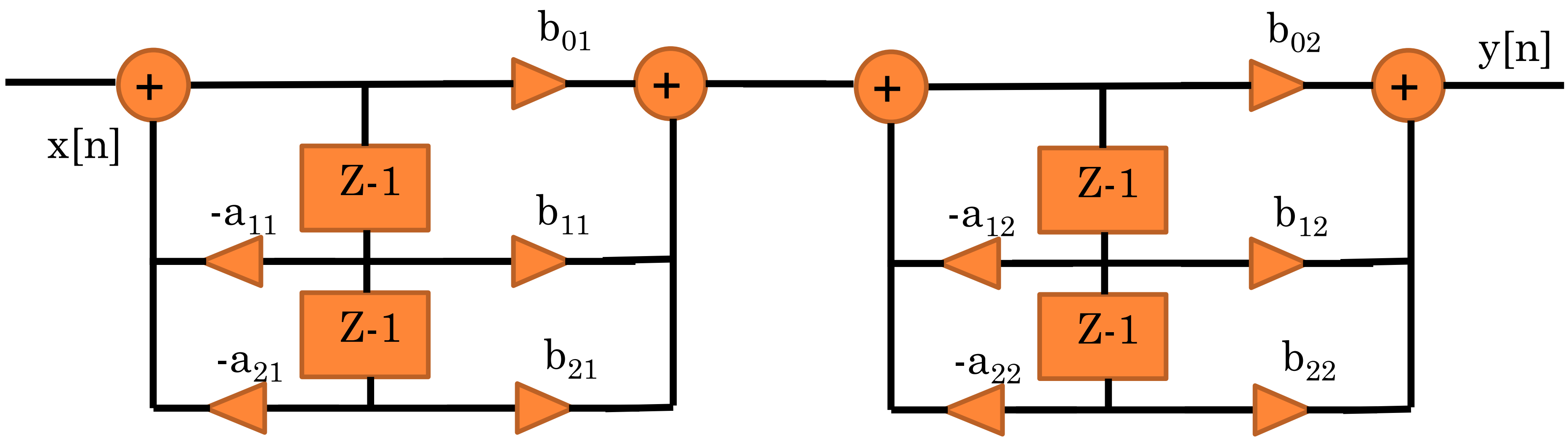


Actually, for H1(z), all 4 roots are at z = -0.9 Numerical precision



Example: A 4th order IIR Filter becomes two cascade second order filters

- e.g. Commutation $H = AB = BA$



Stage 1

Stage 2

Cascading second order section!

tf2sos

Convert digital filter transfer function data to second-order sections form

Syntax

```
[sos,g]=tf2sos(b,a)
[sos,g] = tf2sos(b,a,'order')
[sos,g]=tf2sos(b,a,'order','scale')
sos=tf2sos(...)
```

Description

`tf2sos` converts a transfer function representation of a given digital filter to an equivalent second-order section representation.

`[sos,g] = tf2sos(b,a)` finds a matrix `sos` in second-order section form with gain `g` that is equivalent to the digital filter represented by transfer function coefficient vectors `a` and `b`.

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_1 + b_2 z^{-1} + \cdots + b_{n+1} z^{-n}}{a_1 + a_2 z^{-1} + \cdots + a_{m+1} z^{-m}}$$

`sos` is an L-by-6 matrix

$$sos = \begin{bmatrix} b_{01} & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & 1 & a_{12} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{0L} & b_{1L} & b_{2L} & 1 & a_{1L} & a_{2L} \end{bmatrix}$$

whose rows contain the numerator and denominator coefficients b_{ik} and a_{ik} of the second-order sections of $H(z)$.

$$H(z) = g \prod_{k=1}^{L} H_k(z) = g \prod_{k=1}^{L} \frac{b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}}$$

Cascading second order section!

TO REDUCE QUANTIZATION EFFECTS DO NOT USE HIGHER POLYNOMIAL

- $$H1(z) = \frac{z^4}{(z+0.9)^4} = \frac{1}{1+3.6z^{-1}+4.86z^{-2}+2.916z^{-3}+\mathbf{0.6561z^{-4}}}$$

```
>> [SOS1,G] = tf2sos([1],[1 3.6 4.86 2.916 0.6561])

SOS1 =

    1.0000         0         0    1.0000    1.8000    0.8100
    1.0000         0         0    1.0000    1.8000    0.8100
```

Stage 1

Stage 2

'B' (feedforward) coefficients

'A' feedback coefficients

```
>> roots([1 1.8 0.81])

ans =

   -0.9000
   -0.9000
```

The roots of SOS are at -0.9, -0.9

```
>> roots([1 1.8 0.811])

ans =

  -0.9000 + 0.0316i
  -0.9000 - 0.0316i
```

The roots of perturbed SOS are at -0.9 +j0.0316 and -0.9-j0.0316

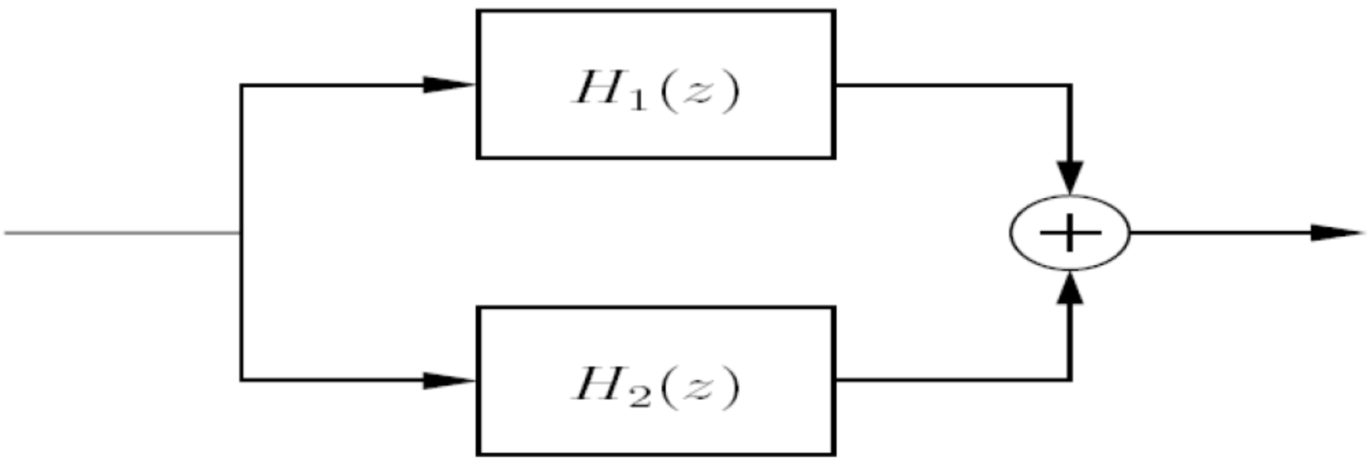
PARALLEL STRUCTURES

- H(z) can be decomposed into partial fractions, in such a case, a parallel structure is formed.

The idea: rewrite $H(z)$ as a sum of filters

$$H(z) = \sum_{i=1}^{K} H_i(z)$$

Parallel Structure for $K = 2$



MATLAB: TO DECOMPOSE INTO PARTIAL FRACTION EXPANSION

residuez

z-transform partial-fraction expansion

Syntax

```
[r,p,k]=residuez(b,a)
[b,a] = residuez(r,p,k)
```

Description

`residuez` converts a discrete time system, expressed as the ratio of two polynomials, to partial fractior expansion, or residue, form. It also converts the partial fraction expansion back to the original polynomial coefficients.

> **Note** Numerically, the partial fraction expansion of a ratio of polynomials is an ill-posed problem. If the denominator polynomial is near a polynomial with multiple roots, then small changes in the data, including roundoff errors, can cause arbitrarily large changes in the resulting poles and residues. You should use state-space (or pole-zero representations instead.

EXAMPLE: CONVERTING TO PARALLEL STRUCTURE

```
[r,p,k] = residuez(B,A)
```

```
[r,p,k] = residuez([1 -1 0.5],[1 0.1 -0.72])
```

```
Transfer function:
  1 - z^-1 + 0.5 z^-2
-------------------------
1 + 0.1 z^-1 - 0.72 z^-2
```

```
r =

    1.4444
    0.2500
```

```
p =

   -0.9000
    0.8000
```

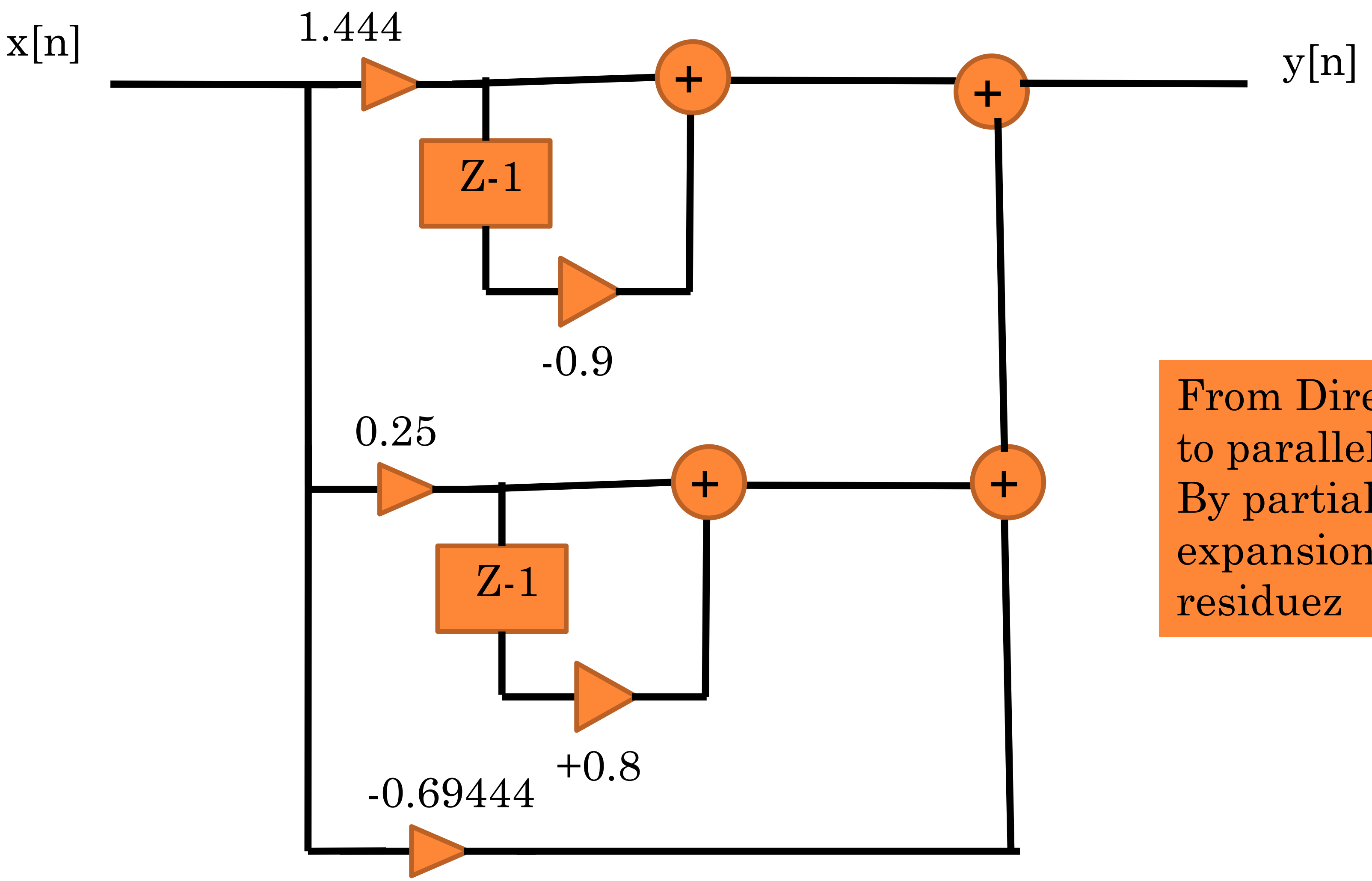
```
k =

   -0.6944
```

$$\frac{B(z)}{A(z)} = \frac{r(1)}{1 - p(1)z^{-1}} + \cdots + \frac{r(n)}{1 - p(n)z^{-1}} + k(1) + k(2)z^{-1} + \cdots + k(m-n+1)z^{-(m-n)}$$

$$\frac{B(z)}{A(z)} = \frac{1 - z^{-1} + 0.5z^{-2}}{1 + 0.1z^{-1} - 0.72z^{-2}} = \frac{1.444}{1 + 0.9z^{-1}} + \frac{0.25}{1 - 0.8z^{-1}} - 0.69444$$

PARALLEL STRUCTURE

$$\frac{B(z)}{A(z)} = \frac{1 - z^{-1} + 0.5z^{-2}}{1 + 0.1z^{-1} - 0.72z^{-2}} = \frac{1.4444}{1 + 0.9z^{-1}} + \frac{0.25}{1 - 0.8z^{-1}} - 0.69444$$



From Direct Form to parallel Form By partial fraction expansion using residuez

Main Characteristics of the Parallel Structure

Simple to implement

Sometimes has an advantage over the cascade realisation in terms of *internally generated quantisation noise*, not much though (no amplication of errors over various stages)

Errors of coefficient quantization of $H_i(z)$) affects zeros of $H(z)$

longer coefficient wordlengths required to ensure stability

Zeros on the unit circle in the overall transfer function are not preserved

no saving of multipliers can be obtained for filters
having such zeros

SUMMARY

1) We did a quick tour of basic filter structure from DF1 to cascade to parallel.
2) We saw that higher order polynomial is sensitive to quantization and hence

   Second order structure is preferred.

3) We used Matlab's tf2sos to get second order cascade structure
4) We used Matlab's command residuez to decompose H(z) into parallel structure.

Things missing:

5) We have not touch on lattice structures, state-space structures.
6) Quantization effects on coefficients and accumulation of errors.