10: Digital Filter Structures

Direct Forms

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

Direct Forms

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

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

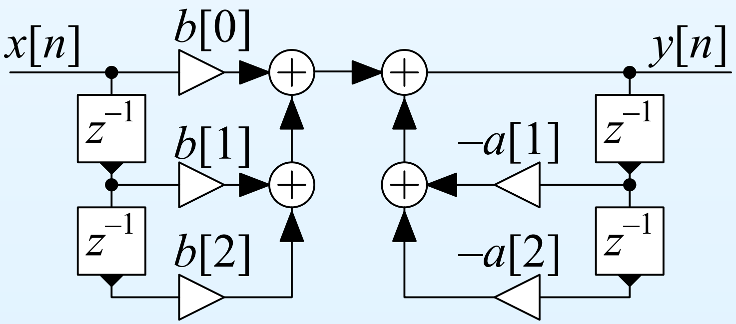
Direct Forms

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 Forms

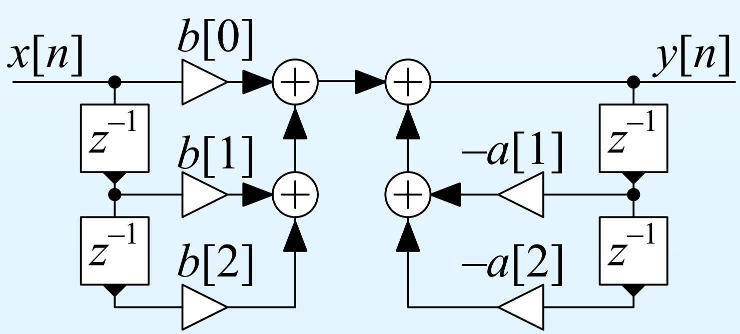
Filter: $H(z) = \frac{B(z)}{A(z)}$ with input $x[n]$ and output $y[n]$

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

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

Direct Form 1:

- Direct implementation of difference equation



Direct Forms

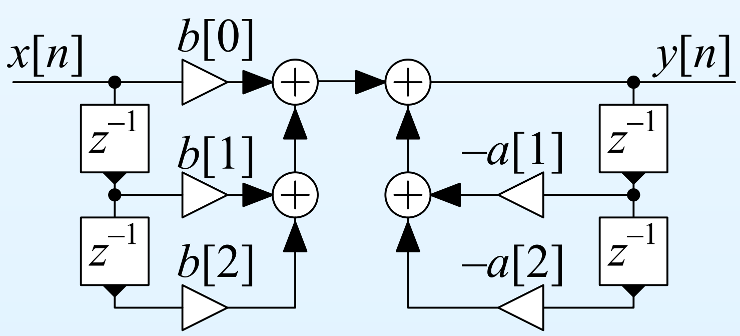
Filter: $H(z) = \frac{B(z)}{A(z)}$ with input $x[n]$ and output $y[n]$

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

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 $\frac{1}{A(z)}$



Direct Forms

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

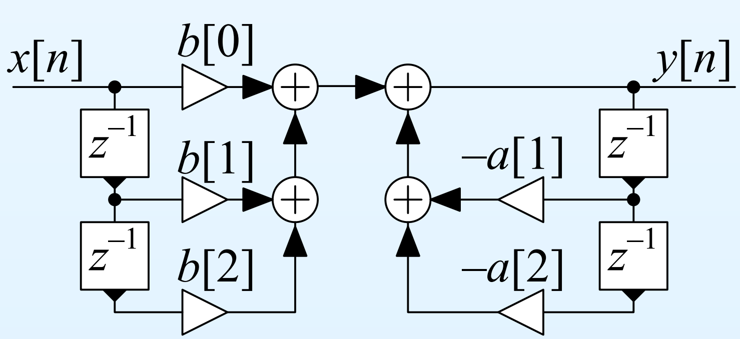
$$y[n] = \sum_{k=0}^{M} b[k]x[n-k] - \sum_{k=1}^{N} a[k]y[n-k]$$

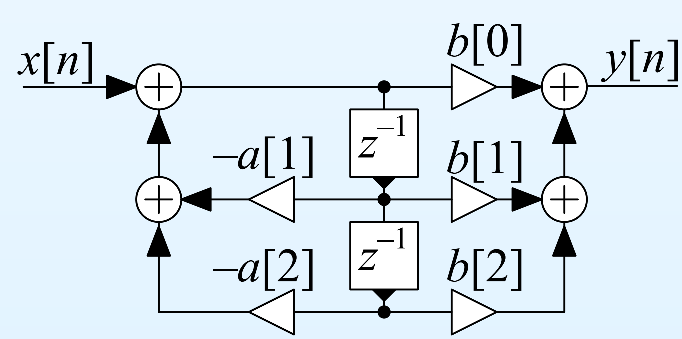
Direct forms use coefficients $a[k]$ and $b[k]$ directly

Direct Form 1:

- Direct implementation of difference equation
- Can view as $B(z)$ followed by $\frac{1}{A(z)}$

Direct Form II:





Direct Forms

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

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

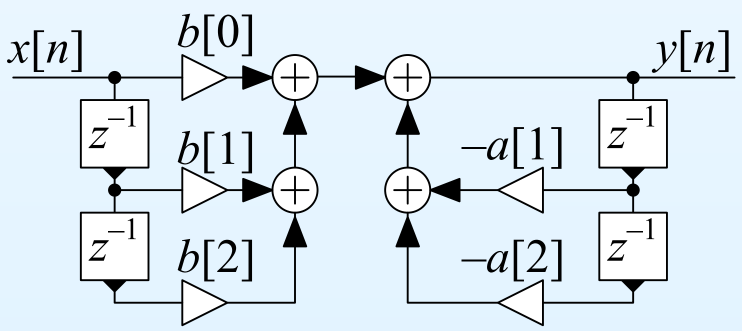
Direct forms use coefficients $a[k]$ and $b[k]$ directly

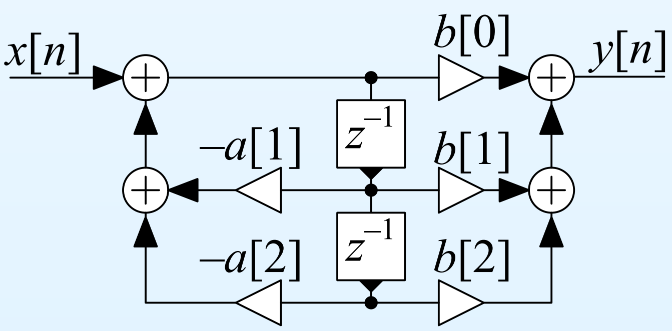
Direct Form 1:

- Direct implementation of difference equation
- Can view as $B(z)$ followed by $\frac{1}{A(z)}$

Direct Form II:

- Implements $\frac{1}{A(z)}$ followed by $B(z)$





Direct Forms

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

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

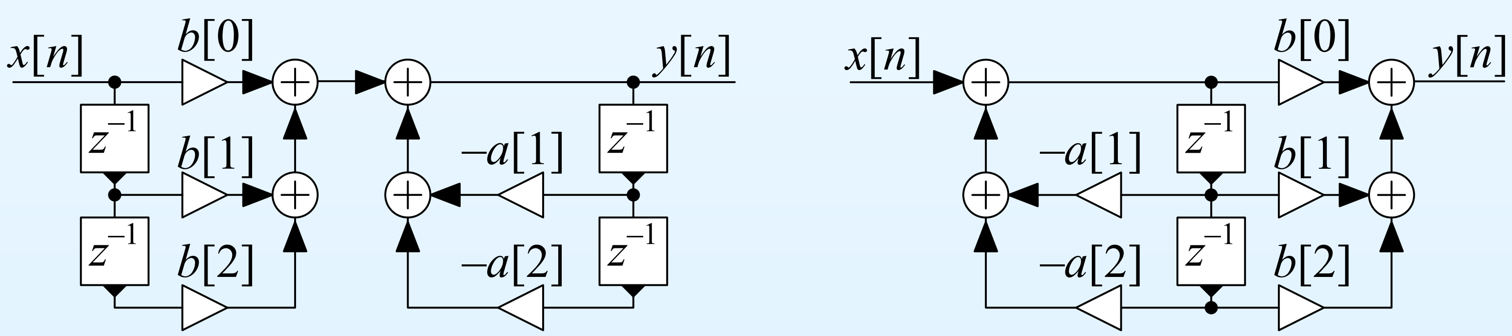
Direct forms use coefficients $a[k]$ and $b[k]$ directly

Direct Form 1:

- Direct implementation of difference equation
- Can view as $B(z)$ followed by $\frac{1}{A(z)}$

Direct Form II:

- Implements $\frac{1}{A(z)}$ followed by $B(z)$
- Saves on delays (= storage)



Transposition

Can convert any block diagram into an equivalent transposed form:

Transposition

Can convert any block diagram into an equivalent transposed form:

- Reverse direction of each interconnection

Transposition

Can convert any block diagram into an equivalent transposed form:

- Reverse direction of each interconnection
- Reverse direction of each multiplier

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

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

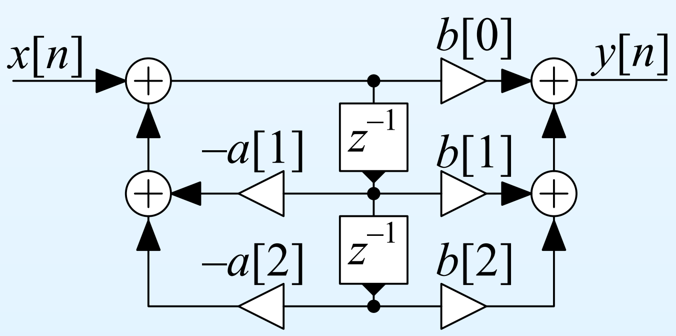
Transposition

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 $\longrightarrow$ Direct Form II$_t$



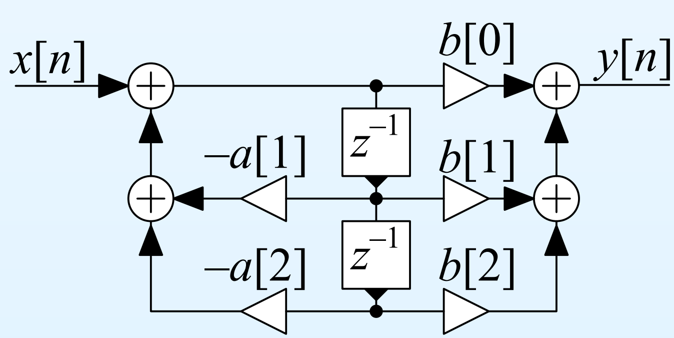
Transposition

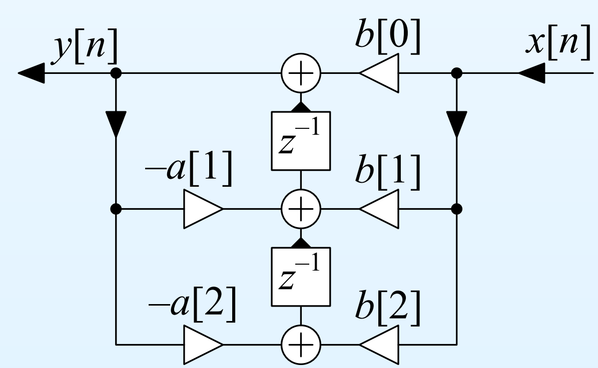
Can convert any block diagram into an equivalent transposed form:

- 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 $\longrightarrow$ Direct Form II$_t$





Transposition

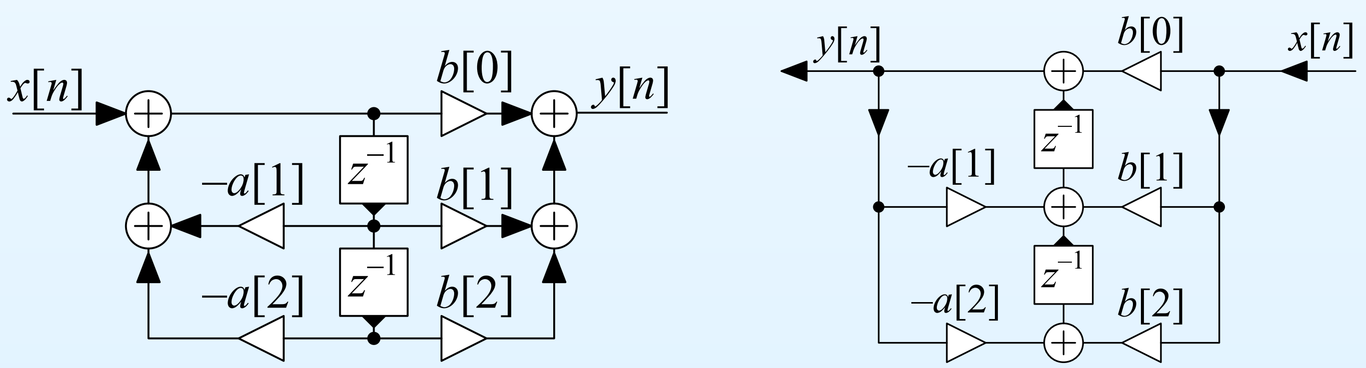
Can convert any block diagram into an equivalent transposed form:

- 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



Transposition

Can convert any block diagram into an equivalent transposed form:

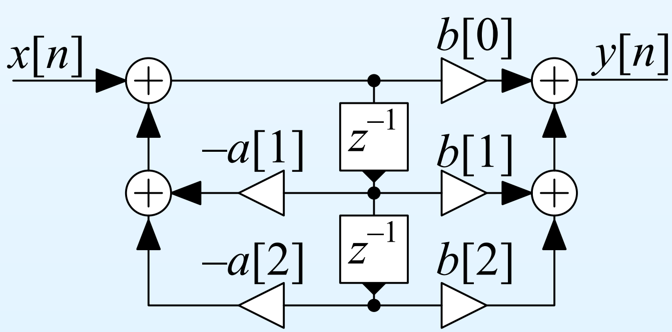
- Reverse direction of each interconnection
- Reverse direction of each multiplier
- Change junctions to adders and vice-versa
- Interchange the input and output signals

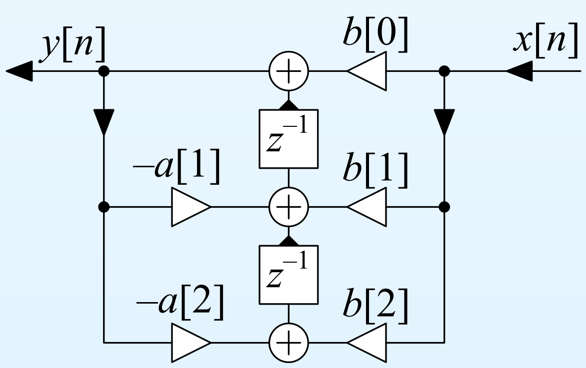
Example:

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).



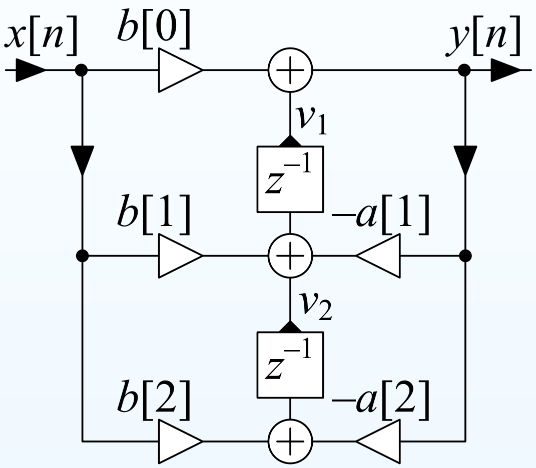


State Space

$\mathbf{v}[n]$ is a vector of delay element outputs

Can write: $\mathbf{v}[n+1] = \mathbf{P}\mathbf{v}[n] + \mathbf{q}x[n]$

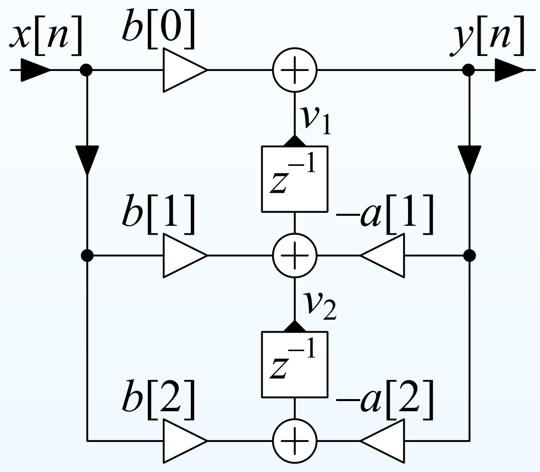
$$y[n] = \mathbf{r}^T\mathbf{v}[n] + sx[n]$$



State Space

$\mathbf{v}[n]$ is a vector of delay element outputs

Can write: $\mathbf{v}[n+1] = \mathbf{P}\mathbf{v}[n] + \mathbf{q}x[n]$
$$y[n] = \mathbf{r}^T\mathbf{v}[n] + sx[n]$$



Example: Direct Form II_t

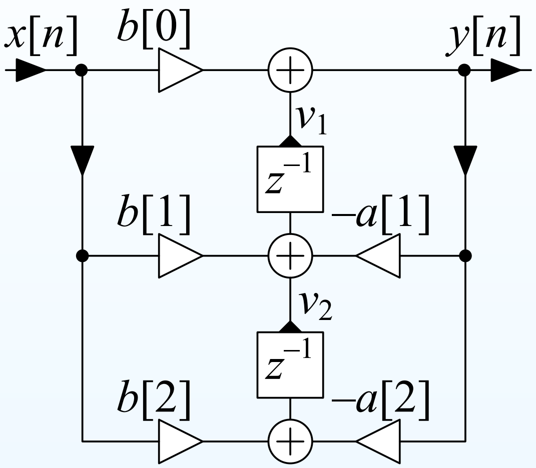
$$\mathbf{P} = \begin{pmatrix} -a[1] & 1 \\ -a[2] & 0 \end{pmatrix}$$

State Space

$\mathbf{v}[n]$ is a vector of delay element outputs

Can write: $\mathbf{v}[n+1] = \mathbf{P}\mathbf{v}[n] + \mathbf{q}x[n]$

$$y[n] = \mathbf{r}^T\mathbf{v}[n] + sx[n]$$



Example: Direct Form II$_t$

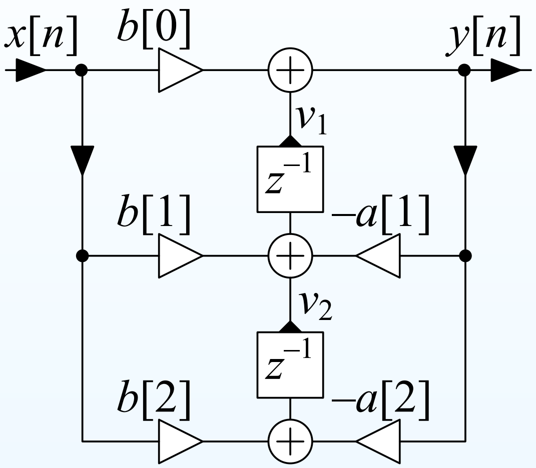
$$\mathbf{P} = \begin{pmatrix} -a[1] & 1 \\ -a[2] & 0 \end{pmatrix} \qquad \mathbf{q} = \begin{pmatrix} b[1] - b[0]a[1] \\ b[2] - b[0]a[2] \end{pmatrix}$$

State Space

$\mathbf{v}[n]$ is a vector of delay element outputs

Can write: $\mathbf{v}[n+1] = \mathbf{P}\mathbf{v}[n] + \mathbf{q}x[n]$

$$y[n] = \mathbf{r}^T\mathbf{v}[n] + sx[n]$$



Example: Direct Form II$_t$

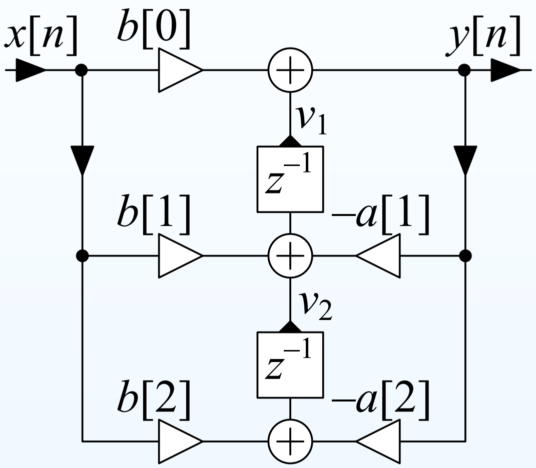
$$\mathbf{P} = \begin{pmatrix} -a[1] & 1 \\ -a[2] & 0 \end{pmatrix} \qquad \mathbf{q} = \begin{pmatrix} b[1] - b[0]a[1] \\ b[2] - b[0]a[2] \end{pmatrix}$$

$$\mathbf{r}^T = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

State Space

$\mathbf{v}[n]$ is a vector of delay element outputs

Can write: $\mathbf{v}[n+1] = \mathbf{P}\mathbf{v}[n] + \mathbf{q}x[n]$
$y[n] = \mathbf{r}^T\mathbf{v}[n] + sx[n]$



Example: Direct Form II$_t$

$$\mathbf{P} = \begin{pmatrix} -a[1] & 1 \\ -a[2] & 0 \end{pmatrix} \qquad \mathbf{q} = \begin{pmatrix} b[1] - b[0]a[1] \\ b[2] - b[0]a[2] \end{pmatrix}$$

$$\mathbf{r}^T = \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad s = b[0]$$

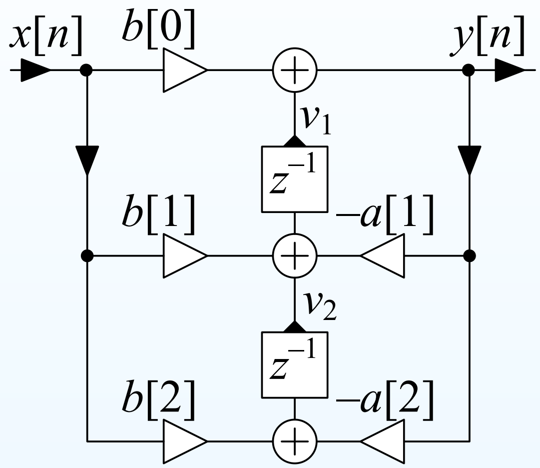
State Space

$\mathbf{v}[n]$ is a vector of delay element outputs

Can write: $\mathbf{v}[n+1] = \mathbf{P}\mathbf{v}[n] + \mathbf{q}x[n]$

$y[n] = \mathbf{r}^T\mathbf{v}[n] + sx[n]$

$\{\mathbf{P}, \mathbf{q}, \mathbf{r}^T, s\}$ is the state-space representation of the filter structure.



Example: Direct Form II$_t$

$$\mathbf{P} = \begin{pmatrix} -a[1] & 1 \\ -a[2] & 0 \end{pmatrix} \qquad \mathbf{q} = \begin{pmatrix} b[1] - b[0]a[1] \\ b[2] - b[0]a[2] \end{pmatrix}$$

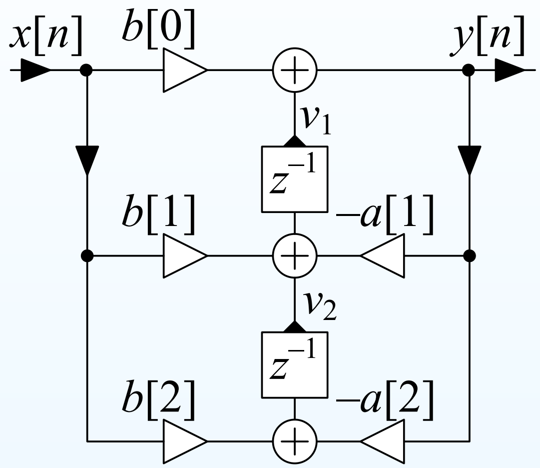
$$\mathbf{r}^T = \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad s = b[0]$$

State Space +

$\mathbf{v}[n]$ is a vector of delay element outputs

Can write: $\mathbf{v}[n+1] = \mathbf{P}\mathbf{v}[n] + \mathbf{q}x[n]$
$$y[n] = \mathbf{r}^T\mathbf{v}[n] + sx[n]$$

$\{\mathbf{P}, \mathbf{q}, \mathbf{r}^T, s\}$ is the state-space representation of the filter structure.



The transfer function is given by:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\det\left(z\mathbf{I}-\mathbf{P}+\mathbf{q}\mathbf{r}^T\right)}{\det(z\mathbf{I}-\mathbf{P})} + s - 1$$

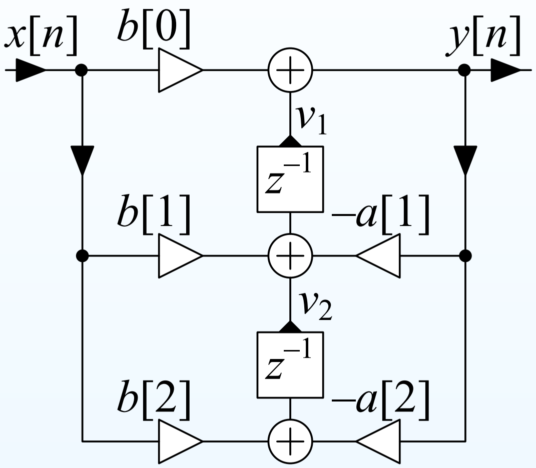
Example: Direct Form II$_t$

$$\mathbf{P} = \begin{pmatrix} -a[1] & 1 \\ -a[2] & 0 \end{pmatrix} \qquad \mathbf{q} = \begin{pmatrix} b[1] - b[0]a[1] \\ b[2] - b[0]a[2] \end{pmatrix}$$

$$\mathbf{r}^T = \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad s = b[0]$$

State Space

$\mathbf{v}[n]$ is a vector of delay element outputs



Can write: $\mathbf{v}[n+1] = \mathbf{P}\mathbf{v}[n] + \mathbf{q}x[n]$
$y[n] = \mathbf{r}^T\mathbf{v}[n] + sx[n]$

$\left\{\mathbf{P}, \mathbf{q}, \mathbf{r}^T, s\right\}$ is the state-space representation of the filter structure.

The transfer function is given by:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\det\left(z\mathbf{I}-\mathbf{P}+\mathbf{q}\mathbf{r}^T\right)}{\det(z\mathbf{I}-\mathbf{P})} + s - 1$$

Example: Direct Form II$_t$

$$\mathbf{P} = \begin{pmatrix} -a[1] & 1 \\ -a[2] & 0 \end{pmatrix} \qquad \mathbf{q} = \begin{pmatrix} b[1]-b[0]a[1] \\ b[2]-b[0]a[2] \end{pmatrix}$$

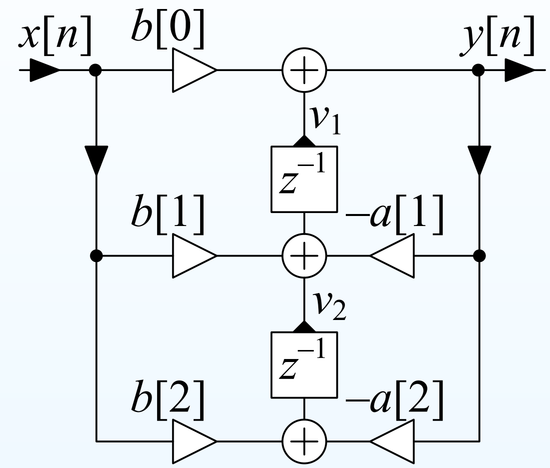
$$\mathbf{r}^T = \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad s = b[0]$$

From which $H(z) = \frac{b[0]z^2+b[1]z+b[2]}{z^2+a[1]z+a[2]}$

State Space

+

$\mathbf{v}[n]$ is a vector of delay element outputs



Can write: $\mathbf{v}[n+1] = \mathbf{P}\mathbf{v}[n] + \mathbf{q}x[n]$
$y[n] = \mathbf{r}^T\mathbf{v}[n] + sx[n]$

$\{\mathbf{P}, \mathbf{q}, \mathbf{r}^T, s\}$ is the state-space representation of the filter structure.

The transfer function is given by:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\det\left(z\mathbf{I}-\mathbf{P}+\mathbf{q}\mathbf{r}^T\right)}{\det(z\mathbf{I}-\mathbf{P})} + s - 1$$

The transposed form has $\mathbf{P} \to \mathbf{P}^T$ and $\mathbf{q} \leftrightarrow \mathbf{r} \quad \Rightarrow \quad$ same $H(z)$

Example: Direct Form II$_t$

$$\mathbf{P} = \begin{pmatrix} -a[1] & 1 \\ -a[2] & 0 \end{pmatrix} \qquad \mathbf{q} = \begin{pmatrix} b[1] - b[0]a[1] \\ b[2] - b[0]a[2] \end{pmatrix}$$

$$\mathbf{r}^T = \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad s = b[0]$$

From which $H(z) = \frac{b[0]z^2+b[1]z+b[2]}{z^2+a[1]z+a[2]}$

Precision Issues

If all computations were exact, it would not make any difference which of the equivalent structures was used.

Precision Issues

If all computations were exact, it would not make any difference which of the equivalent structures was used. However ...

- Coefficient precision

Precision Issues

If all computations were exact, it would not make any difference which of the equivalent structures was used. However ...

- Coefficient precision
  Coefficients are stored to finite precision and so are not exact.

Precision Issues

If all computations were exact, it would not make any difference which of the equivalent structures was used. However ...

- Coefficient precision
  Coefficients are stored to finite precision and so are not exact. The filter actually implemented is therefore incorrect.

Precision Issues

If all computations were exact, it would not make any difference which of the equivalent structures was used. However ...

- Coefficient precision
  Coefficients are stored to finite precision and so are not exact. The filter actually implemented is therefore incorrect.
- Arithmetic precision
  Arithmetic calculations are not exact.

Precision Issues

If all computations were exact, it would not make any difference which of the equivalent structures was used. However ...

- Coefficient precision
  Coefficients are stored to finite precision and so are not exact. The filter actually implemented is therefore incorrect.
- Arithmetic precision
  Arithmetic calculations are not exact.
  - Worst case for arithmetic errors is when calculating the difference between two similar values:

Precision Issues

If all computations were exact, it would not make any difference which of the equivalent structures was used. However ...

- Coefficient precision
  Coefficients are stored to finite precision and so are not exact. The filter actually implemented is therefore incorrect.
- Arithmetic precision
  Arithmetic calculations are not exact.
  - Worst case for arithmetic errors is when calculating the difference between two similar values:
    $1.23456789 - 1.23455678 = 0.00001111$: 9 s.f. $\rightarrow$ 4 s.f.

Precision Issues

If all computations were exact, it would not make any difference which of the equivalent structures was used. However ...

- **Coefficient precision**
  Coefficients are stored to finite precision and so are not exact.
  The filter actually implemented is therefore incorrect.

- **Arithmetic precision**
  Arithmetic calculations are not exact.
  - Worst case for arithmetic errors is when calculating the difference between two similar values:
    $1.23456789 - 1.23455678 = 0.00001111$: 9 s.f. $\rightarrow$ 4 s.f.

  Arithmetic errors introduce noise that is then filtered by the transfer function between the point of noise creation and the output.

Coefficient Sensitivity

The roots of high order polynomials can be very sensitive to small changes in coefficient values.

Coefficient Sensitivity

The roots of high order polynomials can be very sensitive to small changes in coefficient values.

Wilkinson's polynomial: (famous example)

$$f(x) = \prod_{n=1}^{20} (x-n) = x^{20} - 210x^{19} + 20615x^{18} - \ldots$$

has roots well separated on the real axis.

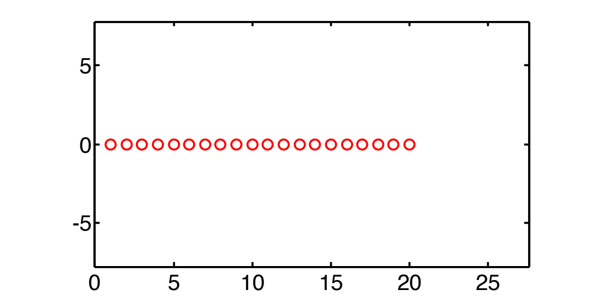
Coefficient Sensitivity

The roots of high order polynomials can be very sensitive to small changes in coefficient values.

Wilkinson's polynomial: (famous example)

$$f(x) = \prod_{n=1}^{20} (x-n) = x^{20} - 210x^{19} + 20615x^{18} - \ldots$$

has roots well separated on the real axis.



Coefficient Sensitivity

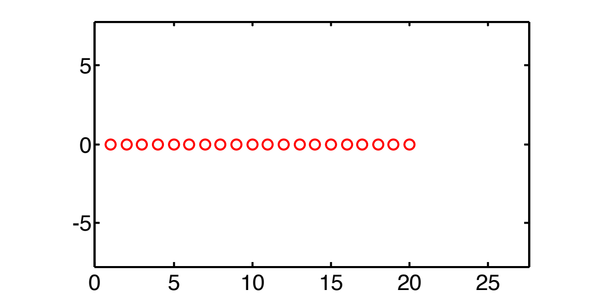
The roots of high order polynomials can be very sensitive to small changes in coefficient values.

Wilkinson's polynomial: (famous example)

$$f(x) = \prod_{n=1}^{20} (x-n) = x^{20} - 210x^{19} + 20615x^{18} - \ldots$$

has roots well separated on the real axis.

Multiplying the coefficient of x^{19} by 1.000001 moves the roots a lot.



Coefficient Sensitivity

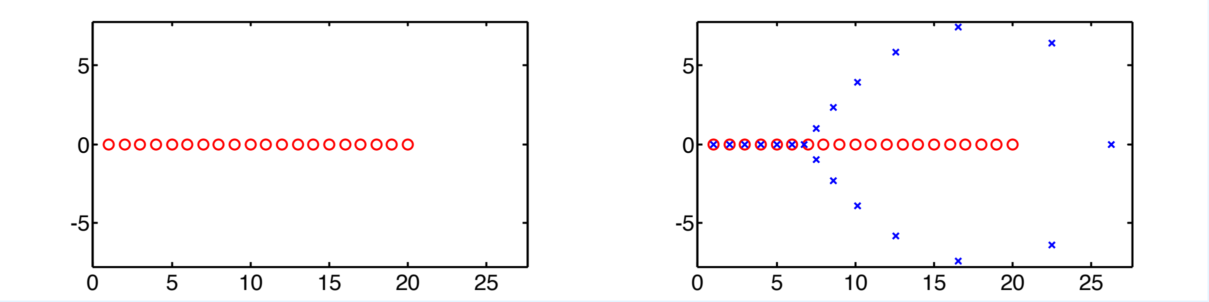
The roots of high order polynomials can be very sensitive to small changes in coefficient values.

Wilkinson's polynomial: (famous example)

$$f(x) = \prod_{n=1}^{20} (x-n) = x^{20} - 210x^{19} + 20615x^{18} - \ldots$$

has roots well separated on the real axis.

Multiplying the coefficient of x^{19} by 1.000001 moves the roots a lot.



Coefficient Sensitivity

The roots of high order polynomials can be very sensitive to small changes in coefficient values.

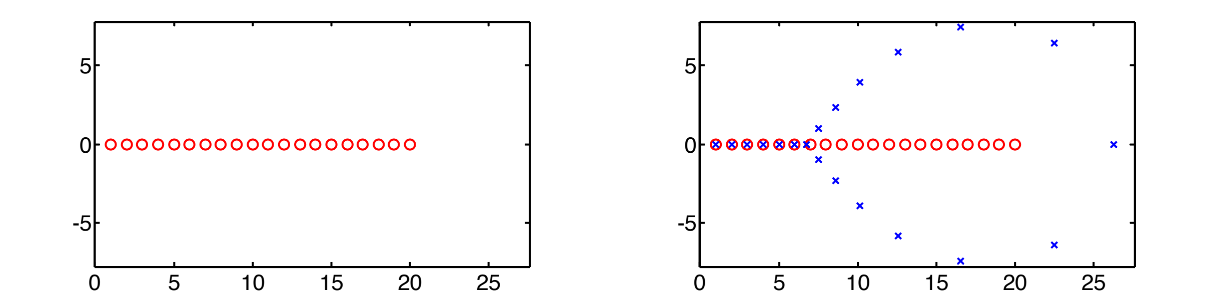
Wilkinson's polynomial: (famous example)

$$f(x) = \prod_{n=1}^{20} (x - n) = x^{20} - 210x^{19} + 20615x^{18} - \ldots$$

has roots well separated on the real axis.

Multiplying the coefficient of x^{19} by 1.000001 moves the roots a lot.

"Speaking for myself I regard it as the most traumatic experience in my career as a numerical analyst", James Wilkinson 1984



Coefficient Sensitivity

The roots of high order polynomials can be very sensitive to small changes in coefficient values.

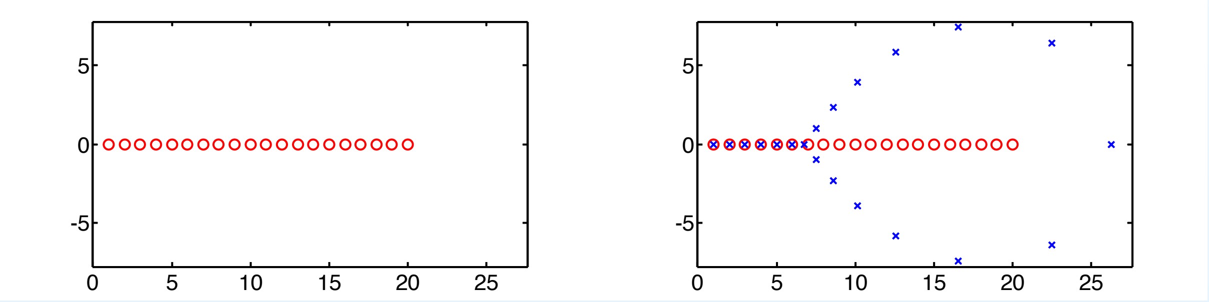
Wilkinson's polynomial: (famous example)

$$f(x) = \prod_{n=1}^{20} (x - n) = x^{20} - 210x^{19} + 20615x^{18} - \ldots$$

has roots well separated on the real axis.

Multiplying the coefficient of x^{19} by 1.000001 moves the roots a lot.

"Speaking for myself I regard it as the most traumatic experience in my career as a numerical analyst", James Wilkinson 1984



Moral: Avoid using direct form for filters orders over about 10.

Cascaded Biquads

Avoid high order polynomials by factorizing into quadratic terms:

$$\frac{B(z)}{A(z)} = g\frac{\prod\left(1+b_{k,1}z^{-1}+b_{k,2}z^{-2}\right)}{\prod\left(1+a_{k,1}z^{-1}+a_{k,2}z^{-2}\right)}$$

Cascaded Biquads

Avoid high order polynomials by factorizing into quadratic terms:

$$\frac{B(z)}{A(z)} = g\frac{\prod\left(1+b_{k,1}z^{-1}+b_{k,2}z^{-2}\right)}{\prod\left(1+a_{k,1}z^{-1}+a_{k,2}z^{-2}\right)} = g\prod_{k=1}^{K}\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$$

Cascaded Biquads

Avoid high order polynomials by factorizing into quadratic terms:

$$\frac{B(z)}{A(z)} = g\frac{\prod\left(1+b_{k,1}z^{-1}+b_{k,2}z^{-2}\right)}{\prod\left(1+a_{k,1}z^{-1}+a_{k,2}z^{-2}\right)} = g\prod_{k=1}^{K}\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$$

where $K = \max\left(\left\lceil\frac{M}{2}\right\rceil, \left\lceil\frac{N}{2}\right\rceil\right)$.

Cascaded Biquads

Avoid high order polynomials by factorizing into quadratic terms:

$$\frac{B(z)}{A(z)} = g\frac{\prod\left(1+b_{k,1}z^{-1}+b_{k,2}z^{-2}\right)}{\prod\left(1+a_{k,1}z^{-1}+a_{k,2}z^{-2}\right)} = g\prod_{k=1}^{K}\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$$

where $K = \max\left(\left\lceil\frac{M}{2}\right\rceil, \left\lceil\frac{N}{2}\right\rceil\right)$.

The term $\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$ is a biquad (bi-quadratic section).

Cascaded Biquads

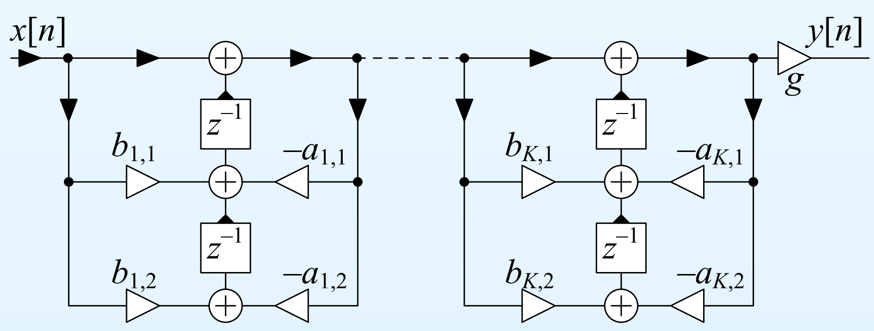
Avoid high order polynomials by factorizing into quadratic terms:

$$\frac{B(z)}{A(z)} = g\frac{\prod\left(1+b_{k,1}z^{-1}+b_{k,2}z^{-2}\right)}{\prod\left(1+a_{k,1}z^{-1}+a_{k,2}z^{-2}\right)} = g\prod_{k=1}^{K}\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$$

where $K = \max\left(\left\lceil \frac{M}{2}\right\rceil, \left\lceil \frac{N}{2}\right\rceil\right)$.

The term $\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$ is a biquad (bi-quadratic section).

Direct Form II Transposed



Cascaded Biquads

Avoid high order polynomials by factorizing into quadratic terms:

$$\frac{B(z)}{A(z)} = g\frac{\prod\left(1+b_{k,1}z^{-1}+b_{k,2}z^{-2}\right)}{\prod\left(1+a_{k,1}z^{-1}+a_{k,2}z^{-2}\right)} = g\prod_{k=1}^{K}\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$$

where $K = \max\left(\left\lceil \frac{M}{2}\right\rceil, \left\lceil \frac{N}{2}\right\rceil\right)$.

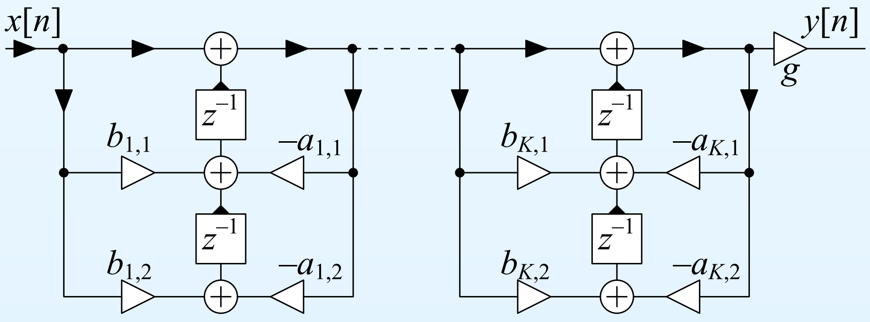
The term $\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$ is a biquad (bi-quadratic section).

We need to choose:

(a) which poles to pair with which zeros in each biquad

Direct Form II

Transposed



Cascaded Biquads

Avoid high order polynomials by factorizing into quadratic terms:

$$\frac{B(z)}{A(z)} = g\frac{\prod\left(1+b_{k,1}z^{-1}+b_{k,2}z^{-2}\right)}{\prod\left(1+a_{k,1}z^{-1}+a_{k,2}z^{-2}\right)} = g\prod_{k=1}^{K}\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$$

where $K = \max\left(\left\lceil \frac{M}{2}\right\rceil, \left\lceil \frac{N}{2}\right\rceil\right)$.

The term $\frac{1+b_{k,1}z^{-1}+b_{k,2}z^{-2}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}$ is a biquad (bi-quadratic section).

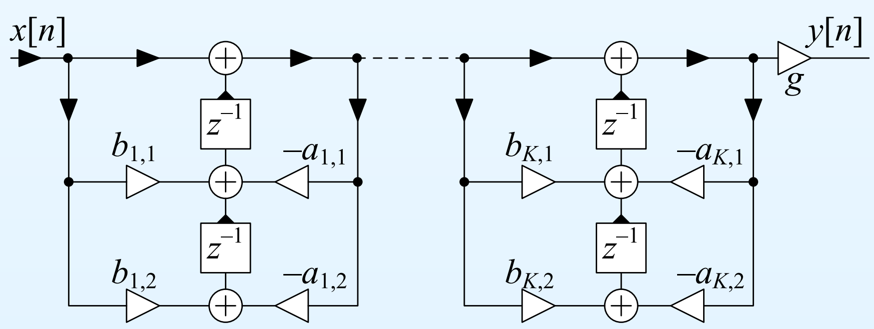
We need to choose:

(a) which poles to pair with which zeros in each biquad

(b) how to order the biquads

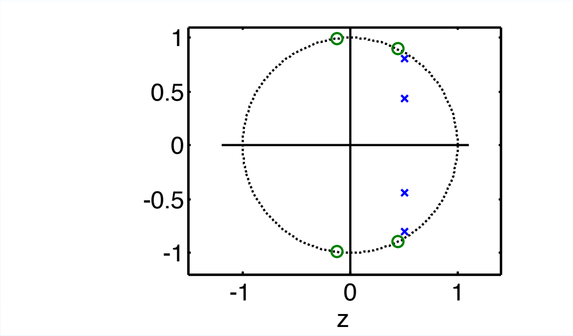
Direct Form II

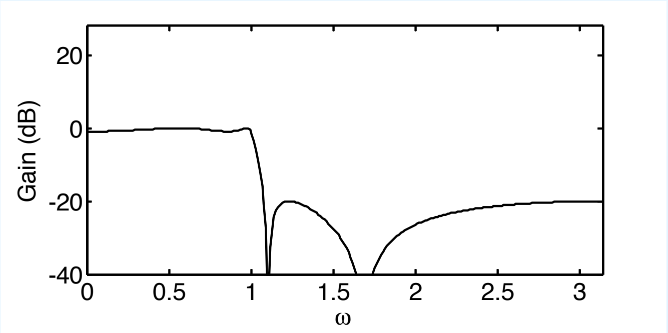
Transposed



Pole-zero Pairing/Ordering

Example: Elliptic lowpass filter

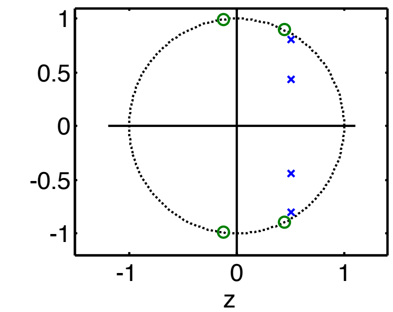


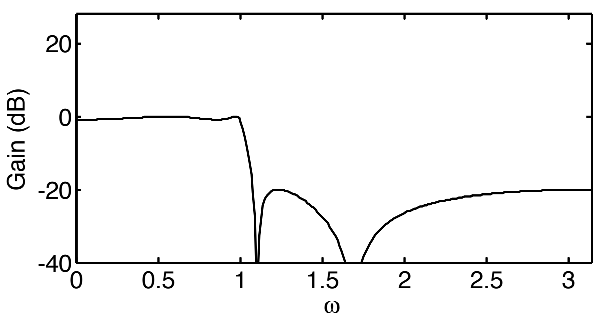


Pole-zero Pairing/Ordering

Example: Elliptic lowpass filter

2 pole pairs and 2 zero pairs

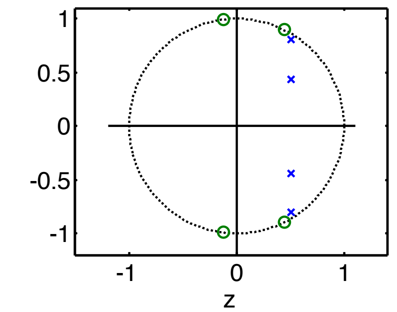


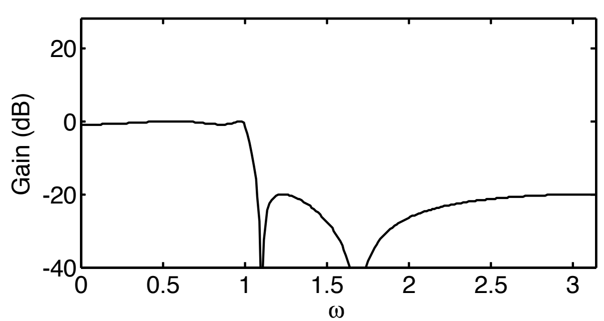


Pole-zero Pairing/Ordering

Example: Elliptic lowpass filter

2 pole pairs and 2 zero pairs
need 2 biquads

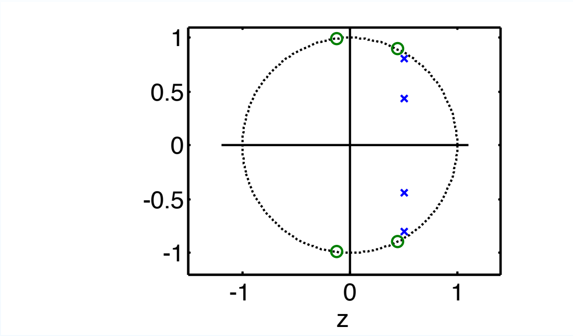




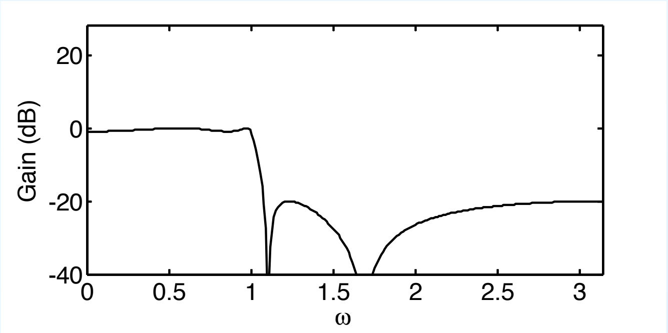
Pole-zero Pairing/Ordering

Example: Elliptic lowpass filter

2 pole pairs and 2 zero pairs
need 2 biquads



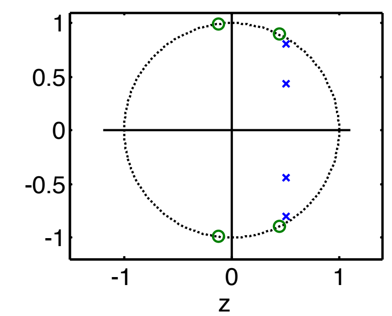
Noise introduced in one biquad is amplified by all the subsequent ones:



Pole-zero Pairing/Ordering

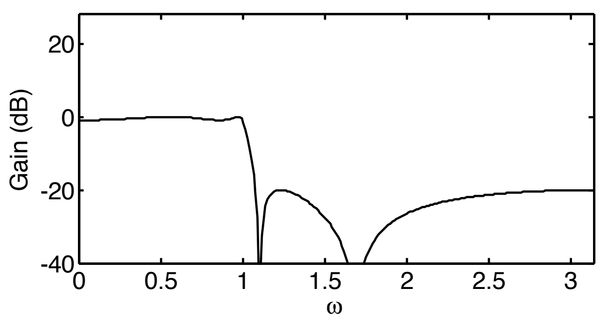
Example: Elliptic lowpass filter

2 pole pairs and 2 zero pairs
need 2 biquads



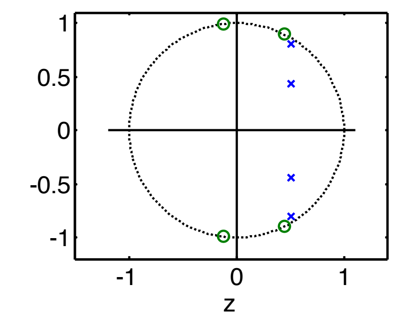
Noise introduced in one biquad is amplified by all the subsequent ones:

- Make the peak gain of each biquad as small as possible



Pole-zero Pairing/Ordering

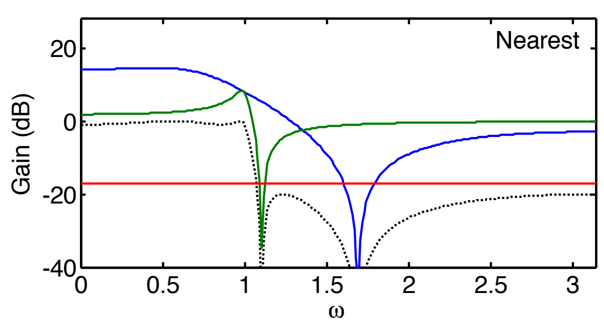
Example: Elliptic lowpass filter



2 pole pairs and 2 zero pairs
need 2 biquads

Noise introduced in one biquad is amplified by all the subsequent ones:

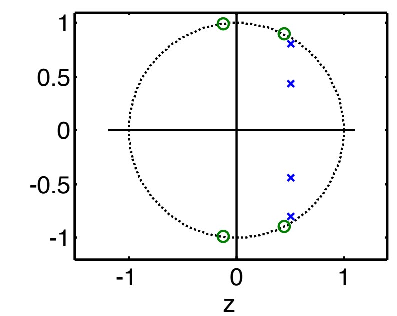
- Make the peak gain of each biquad as small as possible
  - Pair poles with nearest zeros to get lowest peak gain



Pole-zero Pairing/Ordering

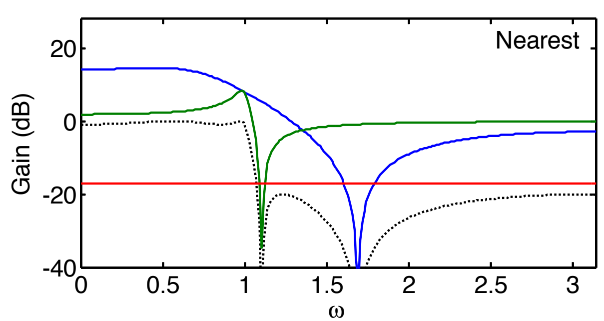
Example: Elliptic lowpass filter

2 pole pairs and 2 zero pairs
need 2 biquads



Noise introduced in one biquad is amplified by all the subsequent ones:

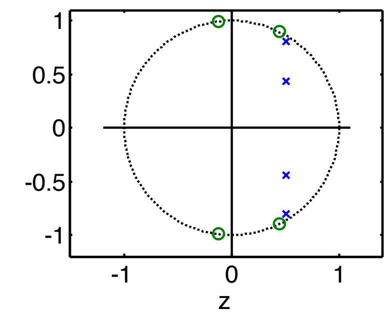
- Make the peak gain of each biquad as small as possible
  - Pair poles with nearest zeros to get lowest peak gain
    begin with the pole nearest the unit circle



Pole-zero Pairing/Ordering

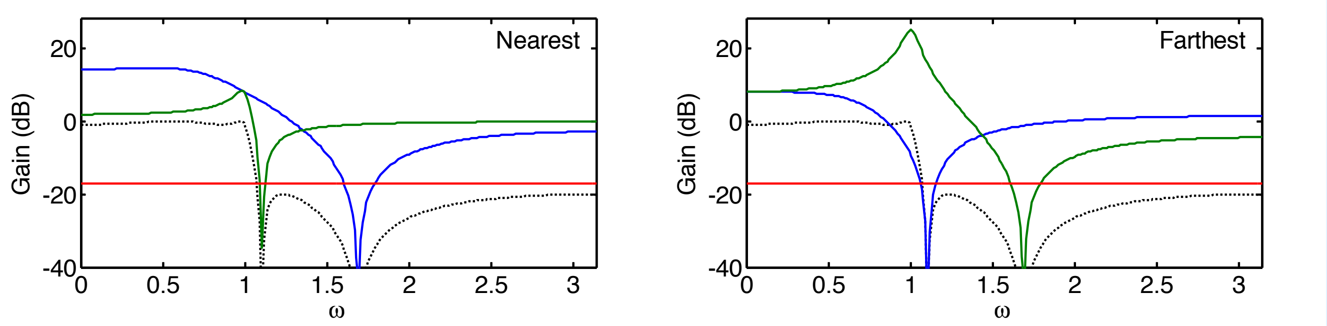
Example: Elliptic lowpass filter

2 pole pairs and 2 zero pairs
need 2 biquads



Noise introduced in one biquad is amplified by all the subsequent ones:

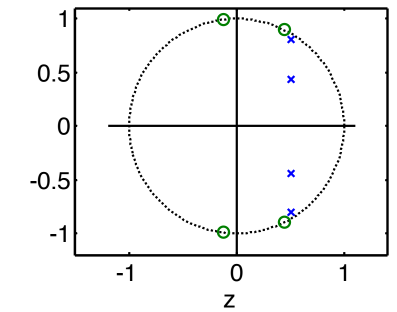
- Make the peak gain of each biquad as small as possible
  - Pair poles with nearest zeros to get lowest peak gain
    begin with the pole nearest the unit circle
  - Pairing with farthest zeros gives higher peak biquad gain



Pole-zero Pairing/Ordering

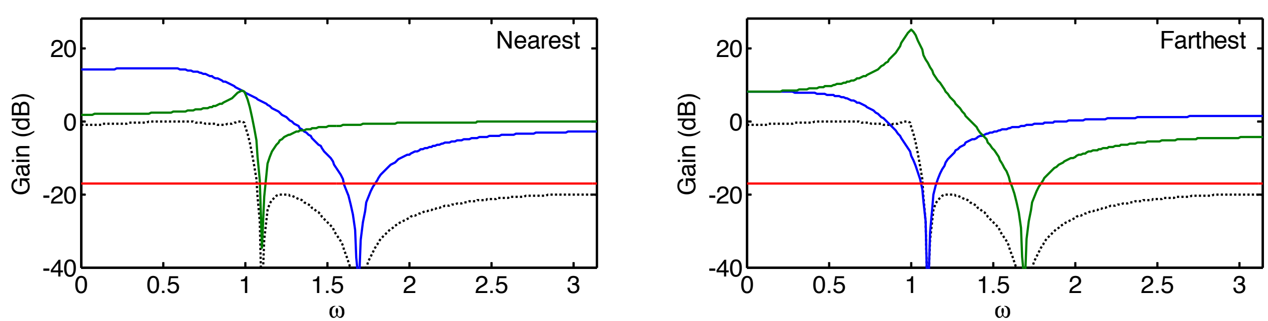
Example: Elliptic lowpass filter

2 pole pairs and 2 zero pairs
need 2 biquads



Noise introduced in one biquad is amplified by all the subsequent ones:

- Make the peak gain of each biquad as small as possible
  - Pair poles with nearest zeros to get lowest peak gain
    begin with the pole nearest the unit circle
  - Pairing with farthest zeros gives higher peak biquad gain
- Poles near the unit circle have the highest peaks and introduce most noise so place them last in the chain



Linear Phase

Implementation can take advantage of any symmetry in the coefficients.

Linear Phase

Implementation can take advantage of any symmetry in the coefficients.

Linear phase filters are always FIR and have symmetric (or, more rarely, antisymmetric) coefficients.

Linear Phase

Implementation can take advantage of any symmetry in the coefficients.

Linear phase filters are always FIR and have symmetric (or, more rarely, antisymmetric) coefficients.

$$H(z) = \sum_{m=0}^{M} h[m] z^{-m} \qquad h[M-m] = h[m]$$

Linear Phase

Implementation can take advantage of any symmetry in the coefficients.

Linear phase filters are always FIR and have symmetric (or, more rarely, antisymmetric) coefficients.

$$H(z) = \sum_{m=0}^{M} h[m]z^{-m} \qquad h[M-m] = h[m]$$

$$= h\left[\frac{M}{2}\right] z^{-\frac{M}{2}} + \sum_{m=0}^{\frac{M}{2}-1} h[m]\left(z^{-m} + z^{m-M}\right) \qquad \text{[m even]}$$

Linear Phase

Implementation can take advantage of any symmetry in the coefficients.

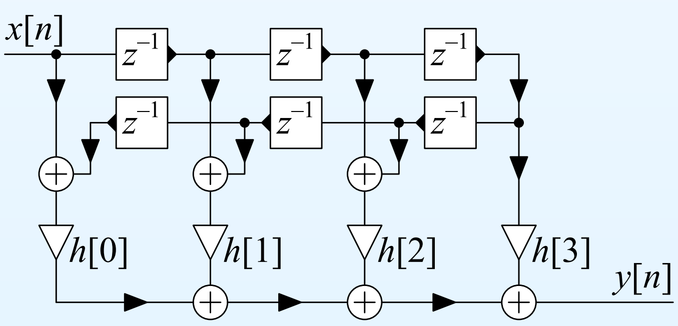
Linear phase filters are always FIR and have symmetric (or, more rarely, antisymmetric) coefficients.

$$H(z) = \sum_{m=0}^{M} h[m] z^{-m} \qquad h[M-m] = h[m]$$

$$= h\left[\frac{M}{2}\right] z^{-\frac{M}{2}} + \sum_{m=0}^{\frac{M}{2}-1} h[m]\left(z^{-m} + z^{m-M}\right) \qquad \text{[m even]}$$

For M even, we only need $\frac{M}{2}+1$ multiplies instead of $M+1$.
We still need M additions and M delays.

$M = 6$:



Linear Phase

Implementation can take advantage of any symmetry in the coefficients.

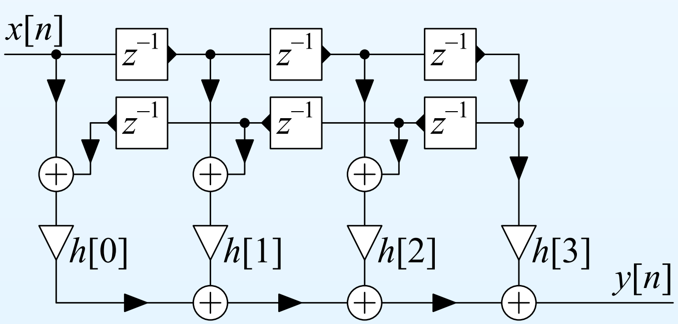
Linear phase filters are always FIR and have symmetric (or, more rarely, antisymmetric) coefficients.

$$H(z) = \sum_{m=0}^{M} h[m] z^{-m} \qquad h[M-m] = h[m]$$

$$= h\left[\frac{M}{2}\right] z^{-\frac{M}{2}} + \sum_{m=0}^{\frac{M}{2}-1} h[m]\left(z^{-m} + z^{m-M}\right) \qquad \text{[m even]}$$

For M even, we only need $\frac{M}{2}+1$ multiplies instead of $M+1$.
We still need M additions and M delays.

$M = 6$:

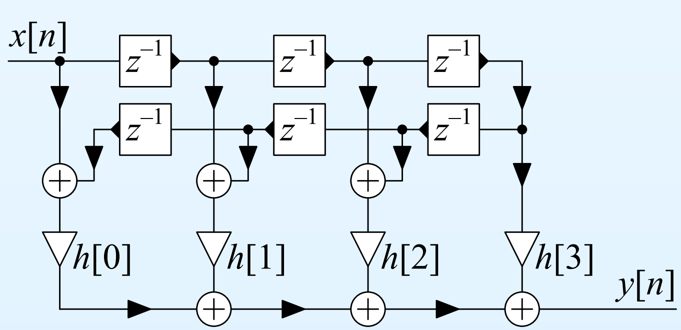


For M odd (no central coefficient), we only need $\frac{M+1}{2}$ multiplies.

Hardware Implementation

Software Implementation:

All that matters is the total number of multiplies and adds



Hardware Implementation

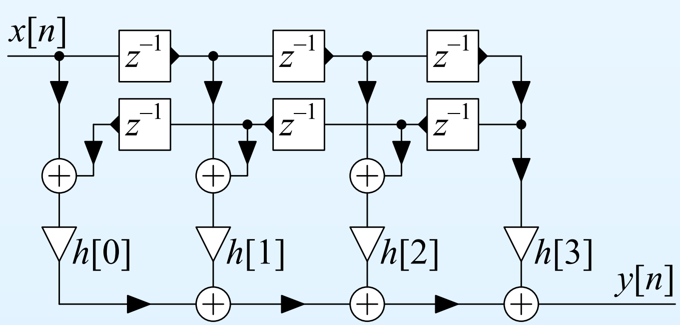
Software Implementation:

All that matters is the total number of multiplies and adds

Hardware Implementation:

Delay elements (z^{-1}) represent storage registers

The maximum clock speed is limited by the number of sequential operations between registers



Hardware Implementation

Software Implementation:

All that matters is the total number of multiplies and adds

Hardware Implementation:

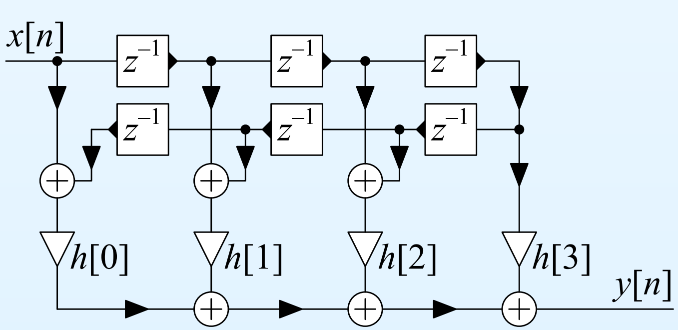
Delay elements (z^{-1}) represent storage registers

The maximum clock speed is limited by the number of sequential operations between registers

Example: Symmetric Linear Phase Filter

Direct form: Maximum sequential delay = $4a + m$

a and m are the delays of adder and multiplier respectively



Hardware Implementation

Software Implementation:

All that matters is the total number of multiplies and adds

Hardware Implementation:

Delay elements (z^{-1}) represent storage registers

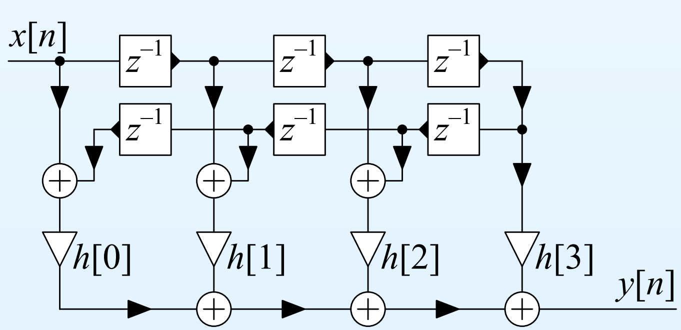
The maximum clock speed is limited by the number of sequential operations between registers

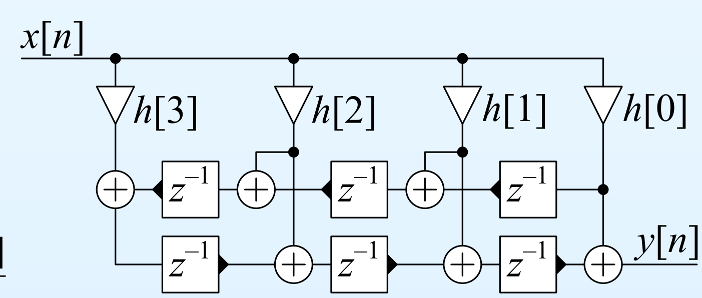
Example: Symmetric Linear Phase Filter

Direct form: Maximum sequential delay = $4a + m$

Transpose form: Maximum sequential delay = $a + m$ ☺

a and m are the delays of adder and multiplier respectively





Allpass Filters

Allpass filters have mirror image numerator and denominator coefficients:

$$b[n] = a[N-n]$$

Allpass Filters

Allpass filters have mirror image numerator and denominator coefficients:

$$b[n] = a[N-n] \qquad \Leftrightarrow \qquad B(z) = z^{-N}A(z^{-1})$$

Allpass Filters

Allpass filters have mirror image numerator and denominator coefficients:

$$b[n] = a[N-n] \quad \Leftrightarrow \quad B(z) = z^{-N} A(z^{-1})$$

$$\Rightarrow \left|H(e^{j\omega})\right| \equiv 1 \forall \omega$$

Allpass Filters

Allpass filters have mirror image numerator and denominator coefficients:

$$b[n] = a[N-n] \quad \Leftrightarrow \quad B(z) = z^{-N} A(z^{-1})$$

$$\Rightarrow \left|H(e^{j\omega})\right| \equiv 1 \forall \omega$$

There are several efficient structures, e.g.

Allpass Filters

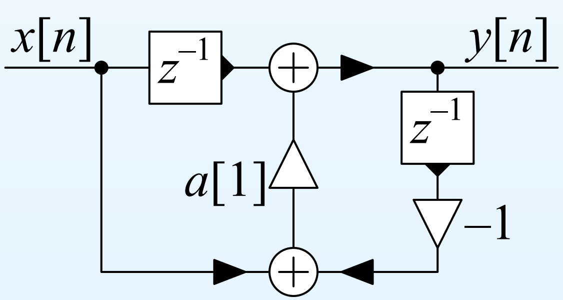
Allpass filters have mirror image numerator and denominator coefficients:

$$b[n] = a[N-n] \quad \Leftrightarrow \quad B(z) = z^{-N}A(z^{-1})$$

$$\Rightarrow \left|H(e^{j\omega})\right| \equiv 1 \forall \omega$$

There are several efficient structures, e.g.

- **First Order:** $H(z) = \frac{a[1]+z^{-1}}{1+a[1]z^{-1}}$



Allpass Filters

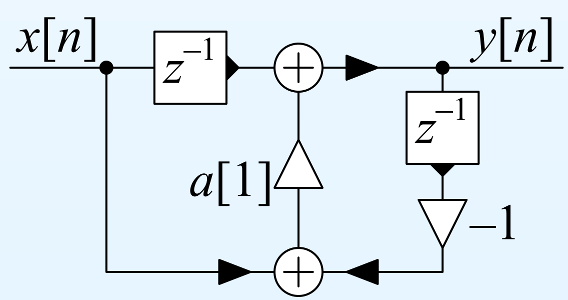
Allpass filters have mirror image numerator and denominator coefficients:

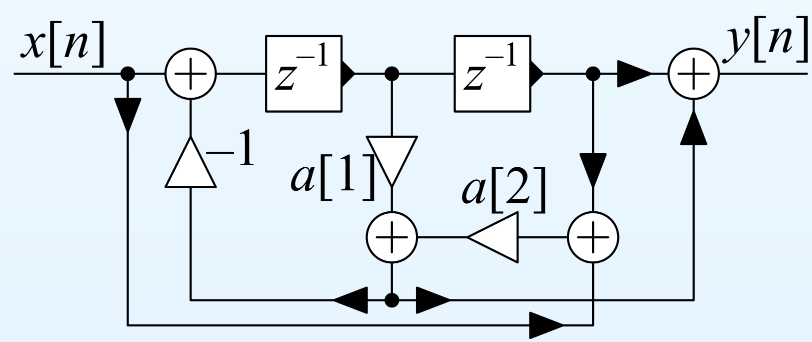
$$b[n] = a[N-n] \quad \Leftrightarrow \quad B(z) = z^{-N}A(z^{-1})$$

$$\Rightarrow \left|H(e^{j\omega})\right| \equiv 1 \forall\omega$$

There are several efficient structures, e.g.

- **First Order:** $H(z) = \frac{a[1]+z^{-1}}{1+a[1]z^{-1}}$
- **Second Order:** $H(z) = \frac{a[2]+a[1]z^{-1}+z^{-2}}{1+a[1]z^{-1}+a[2]z^{-2}}$





Allpass Filters

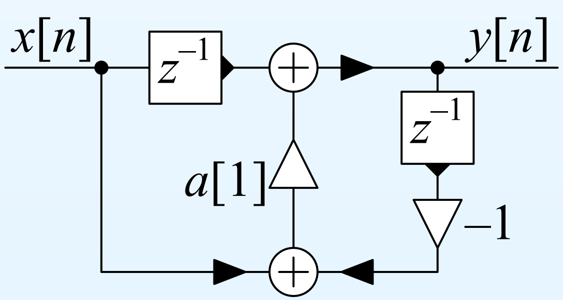
Allpass filters have mirror image numerator and denominator coefficients:

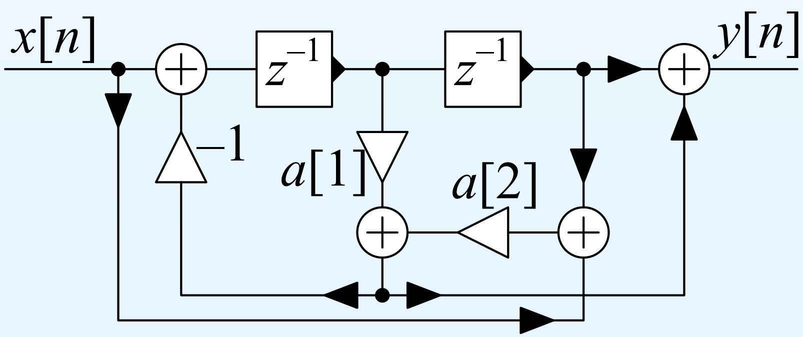
$$b[n] = a[N-n] \quad \Leftrightarrow \quad B(z) = z^{-N}A(z^{-1})$$

$$\Rightarrow \left|H(e^{j\omega})\right| \equiv 1 \forall\omega$$

There are several efficient structures, e.g.

- First Order: $H(z) = \frac{a[1]+z^{-1}}{1+a[1]z^{-1}}$
- Second Order: $H(z) = \frac{a[2]+a[1]z^{-1}+z^{-2}}{1+a[1]z^{-1}+a[2]z^{-2}}$



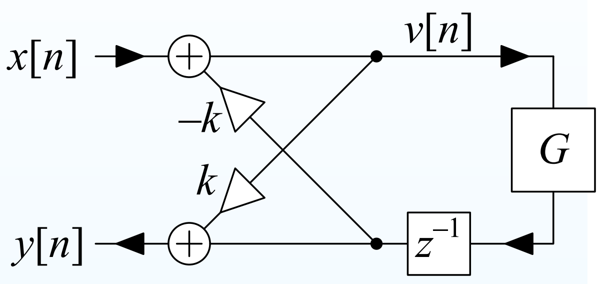


Allpass filters have a gain magnitude of 1 even with coefficient errors.

Lattice Stage

+

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$

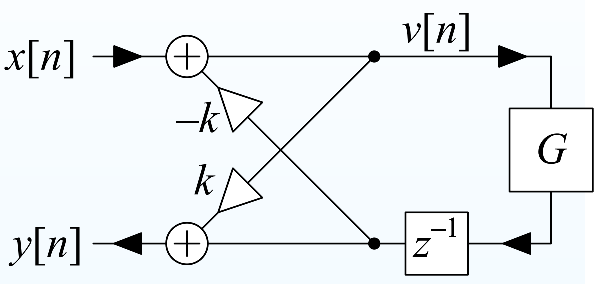


Lattice Stage

+

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$

$$V(z) = X(z) - kGz^{-1}V(z)$$



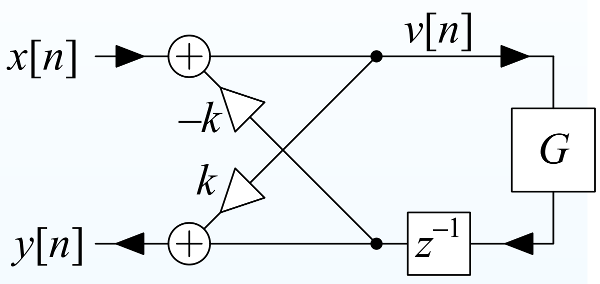
Lattice Stage

+

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$

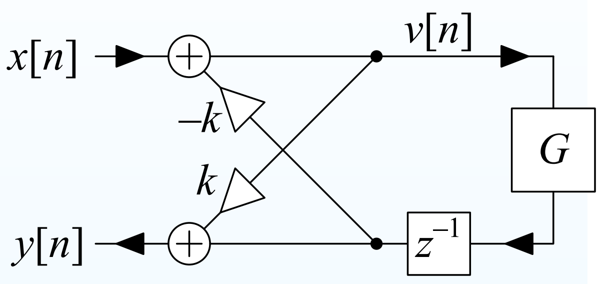
$$V(z) = X(z) - kGz^{-1}V(z)$$

$$\Rightarrow V(z) = \frac{1}{1+kGz^{-1}}X(z)$$



Lattice Stage

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$



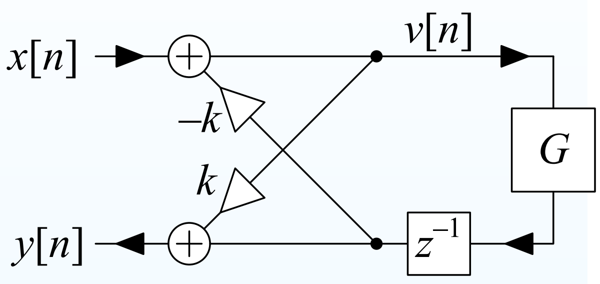
$$V(z) = X(z) - kGz^{-1}V(z)$$

$$\Rightarrow V(z) = \frac{1}{1+kGz^{-1}}X(z)$$

$$Y(z) = kV(z) + Gz^{-1}V(z)$$

Lattice Stage

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$



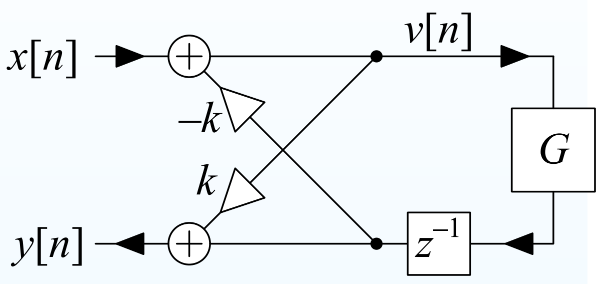
$$V(z) = X(z) - kGz^{-1}V(z)$$

$$\Rightarrow V(z) = \frac{1}{1+kGz^{-1}}X(z)$$

$$Y(z) = kV(z) + Gz^{-1}V(z) = \frac{k+z^{-1}G}{1+kGz^{-1}}X(z)$$

Lattice Stage

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$



$$V(z) = X(z) - kGz^{-1}V(z)$$

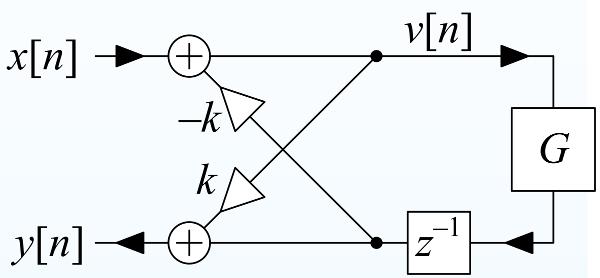
$$\Rightarrow V(z) = \frac{1}{1+kGz^{-1}}X(z)$$

$$Y(z) = kV(z) + Gz^{-1}V(z) = \frac{k+z^{-1}G}{1+kGz^{-1}}X(z)$$

$$\frac{Y(z)}{X(z)} = \frac{kA(z)+z^{-N-1}A(z^{-1})}{A(z)+kz^{-N-1}A(z^{-1})}$$

Lattice Stage

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$



$$V(z) = X(z) - kGz^{-1}V(z)$$

$$\Rightarrow V(z) = \frac{1}{1+kGz^{-1}}X(z)$$

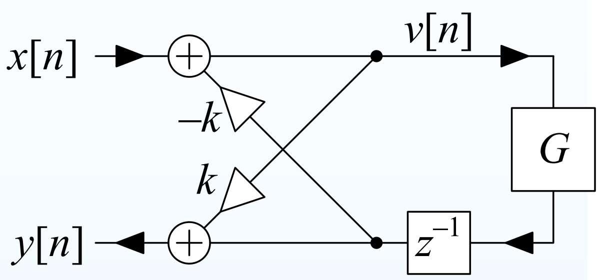
$$Y(z) = kV(z) + Gz^{-1}V(z) = \frac{k+z^{-1}G}{1+kGz^{-1}}X(z)$$

$$\frac{Y(z)}{X(z)} = \frac{kA(z)+z^{-N-1}A(z^{-1})}{A(z)+kz^{-N-1}A(z^{-1})} \triangleq \frac{z^{-(N+1)}D(z^{-1})}{D(z)}$$

Lattice Stage

+

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$



$$V(z) = X(z) - kGz^{-1}V(z)$$

$$\Rightarrow V(z) = \frac{1}{1+kGz^{-1}}X(z)$$

$$Y(z) = kV(z) + Gz^{-1}V(z) = \frac{k+z^{-1}G}{1+kGz^{-1}}X(z)$$

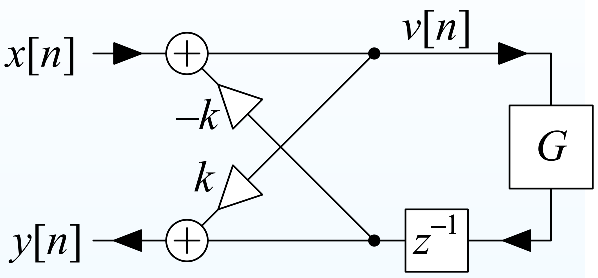
$$\frac{Y(z)}{X(z)} = \frac{kA(z)+z^{-N-1}A(z^{-1})}{A(z)+kz^{-N-1}A(z^{-1})} \triangleq \frac{z^{-(N+1)}D(z^{-1})}{D(z)}$$

Obtaining $\{d[n]\}$ from $\{a[n]\}$:

$$d[n] = \begin{cases} 1 & n = 0 \\ a[n] + ka[N+1-n] & 1 \le n \le N \\ k & n = N+1 \end{cases}$$

Lattice Stage

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$



$$V(z) = X(z) - kGz^{-1}V(z)$$

$$\Rightarrow V(z) = \frac{1}{1+kGz^{-1}}X(z)$$

$$Y(z) = kV(z) + Gz^{-1}V(z) = \frac{k+z^{-1}G}{1+kGz^{-1}}X(z)$$

$$\frac{Y(z)}{X(z)} = \frac{kA(z)+z^{-N-1}A(z^{-1})}{A(z)+kz^{-N-1}A(z^{-1})} \triangleq \frac{z^{-(N+1)}D(z^{-1})}{D(z)}$$

Obtaining $\{d[n]\}$ from $\{a[n]\}$:

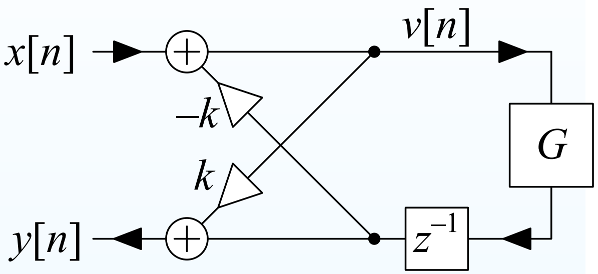
$$d[n] = \begin{cases} 1 & n = 0 \\ a[n] + ka[N+1-n] & 1 \le n \le N \\ k & n = N+1 \end{cases}$$

Obtaining $\{a[n]\}$ from $\{d[n]\}$:

$$k = d[N+1] \qquad a[n] = \frac{d[n]-kd[N+1-n]}{1-k^2}$$

Lattice Stage

Suppose G is allpass: $G(z) = \frac{z^{-N}A(z^{-1})}{A(z)}$



$$V(z) = X(z) - kGz^{-1}V(z)$$

$$\Rightarrow V(z) = \frac{1}{1+kGz^{-1}}X(z)$$

$$Y(z) = kV(z) + Gz^{-1}V(z) = \frac{k+z^{-1}G}{1+kGz^{-1}}X(z)$$

$$\frac{Y(z)}{X(z)} = \frac{kA(z)+z^{-N-1}A(z^{-1})}{A(z)+kz^{-N-1}A(z^{-1})} \triangleq \frac{z^{-(N+1)}D(z^{-1})}{D(z)}$$

Obtaining $\{d[n]\}$ from $\{a[n]\}$:

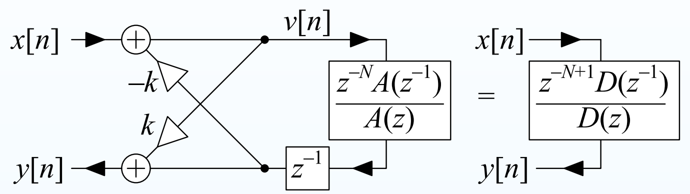
$$d[n] = \begin{cases} 1 & n = 0 \\ a[n] + ka[N+1-n] & 1 \le n \le N \\ k & n = N+1 \end{cases}$$

Obtaining $\{a[n]\}$ from $\{d[n]\}$:

$$k = d[N+1] \qquad a[n] = \frac{d[n]-kd[N+1-n]}{1-k^2}$$

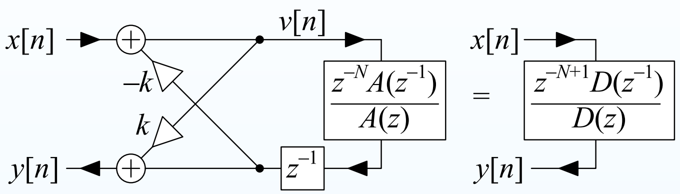
If $G(z)$ is stable then $\frac{Y(z)}{X(z)}$ is stable if and only if $|k| < 1$ (see note)

Suppose $N = 3$, $k = 0.5$ and
$A(z) = 1 + 4z^{-1} - 6z^{-2} + 10z^{-3}$



Example $A(z) \leftrightarrow D(z)$

Suppose $N = 3$, $k = 0.5$ and
$A(z) = 1 + 4z^{-1} - 6z^{-2} + 10z^{-3}$

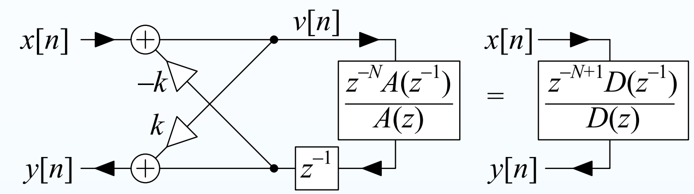


$A(z) \to D(z)$

	z^0	z^{-1}	z^{-2}	z^{-3}	z^{-4}
$A(z)$	1	4	-6	10	
$z^{-4}A(z^{-1})$		10	-6	4	1
$D(z) = A(z) + kz^{-4}A(z^{-1})$	1	9	-9	12	0.5

Example $A(z) \leftrightarrow D(z)$

Suppose $N = 3$, $k = 0.5$ and
$A(z) = 1 + 4z^{-1} - 6z^{-2} + 10z^{-3}$



$A(z) \to D(z)$

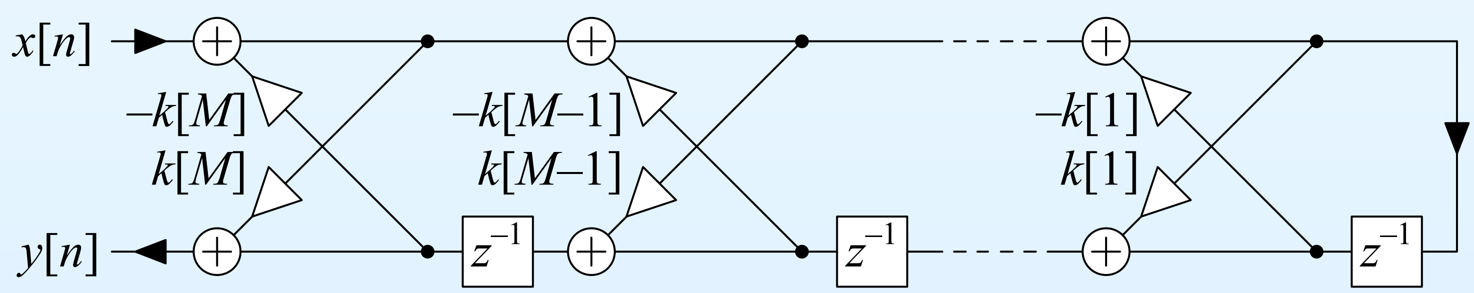
	z^0	z^{-1}	z^{-2}	z^{-3}	z^{-4}
$A(z)$	1	4	-6	10	
$z^{-4}A(z^{-1})$		10	-6	4	1
$D(z) = A(z) + kz^{-4}A(z^{-1})$	1	9	-9	12	0.5

$D(z) \to A(z)$

	z^0	z^{-1}	z^{-2}	z^{-3}	z^{-4}
$D(z)$	1	9	-9	12	0.5
$k = d[N+1]$					0.5
$z^{-4}D(z^{-1})$	0.5	12	-9	9	1
$D(z) - kz^{-4}D(z^{-1})$	0.75	3	-4.5	7.5	0
$A(z) = \frac{D(z) - kz^{-4}D(z^{-1})}{1-k^2}$	1	4	-6	10	0

Allpass Lattice

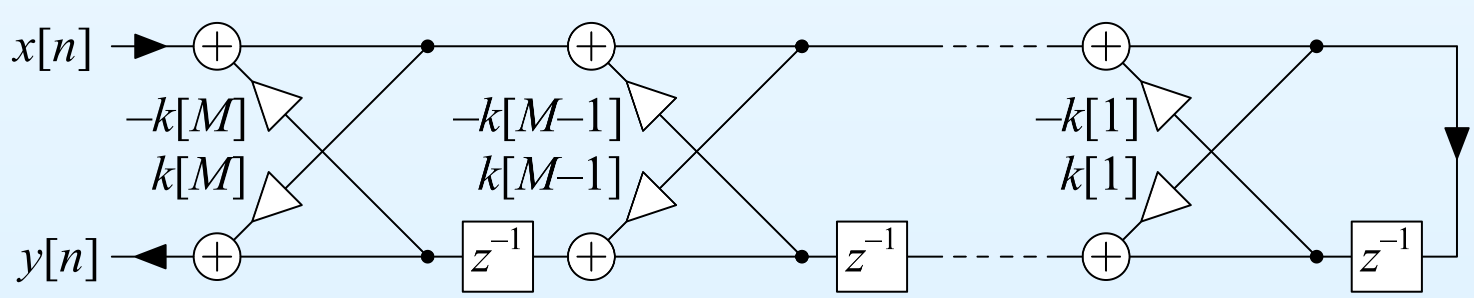
We can implement any allpass filter $H(z) = \frac{z^{-M}A(z^{-1})}{A(z)}$ as a lattice filter with M stages:



Allpass Lattice

We can implement any allpass filter $H(z) = \frac{z^{-M}A(z^{-1})}{A(z)}$ as a lattice filter with M stages:

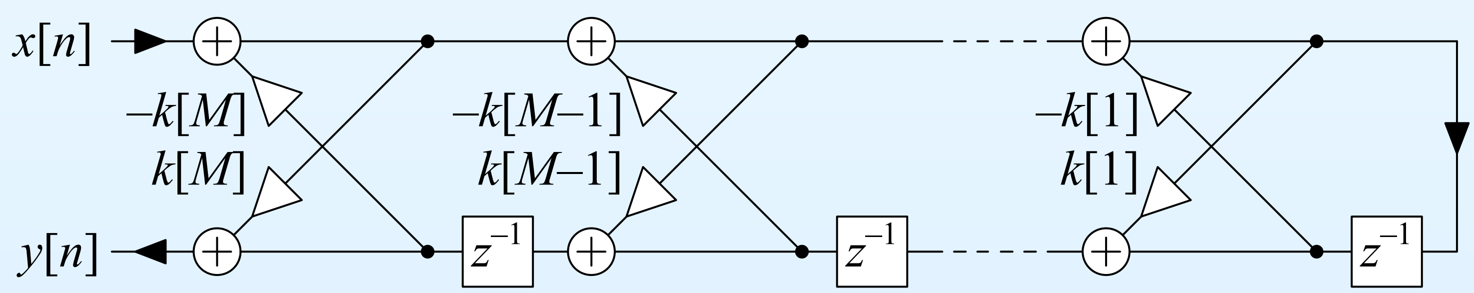
- Initialize $A_M(z) = A(z)$



Allpass Lattice

We can implement any allpass filter $H(z) = \frac{z^{-M}A(z^{-1})}{A(z)}$ as a lattice filter with M stages:

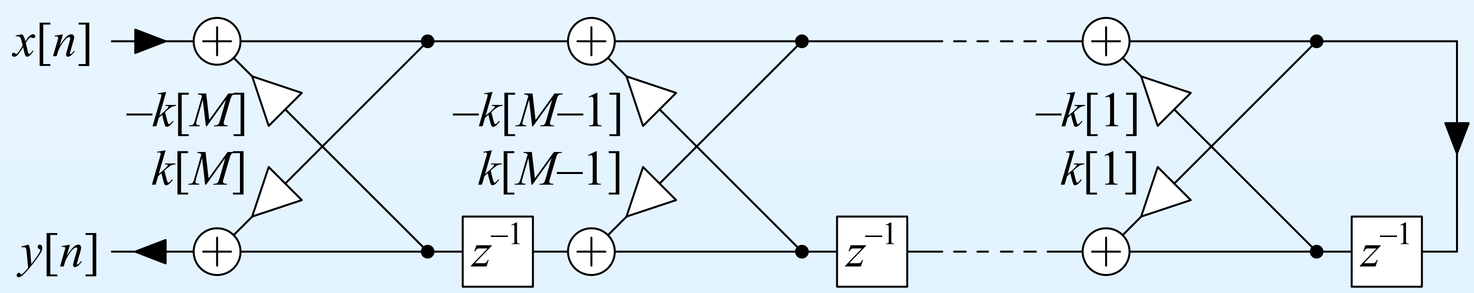
- Initialize $A_M(z) = A(z)$
- Repeat for $m = M : -1 : 1$



Allpass Lattice

We can implement any allpass filter $H(z) = \frac{z^{-M}A(z^{-1})}{A(z)}$ as a lattice filter with M stages:

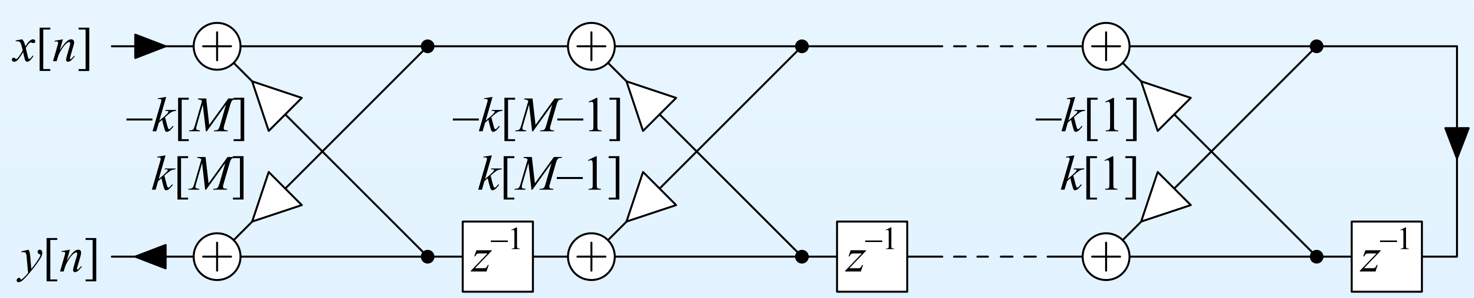
- Initialize $A_M(z) = A(z)$
- Repeat for $m = M : -1 : 1$
    - $k[m] = a_m[m]$



Allpass Lattice

We can implement any allpass filter $H(z) = \frac{z^{-M}A(z^{-1})}{A(z)}$ as a lattice filter with M stages:

- Initialize $A_M(z) = A(z)$
- Repeat for $m = M : -1 : 1$
  - $k[m] = a_m[m]$
  - $a_{m-1}[n] = \frac{a_m[n] - k[m]a_m[m-n]}{1-k^2[m]}$ for $0 \le n \le m-1$

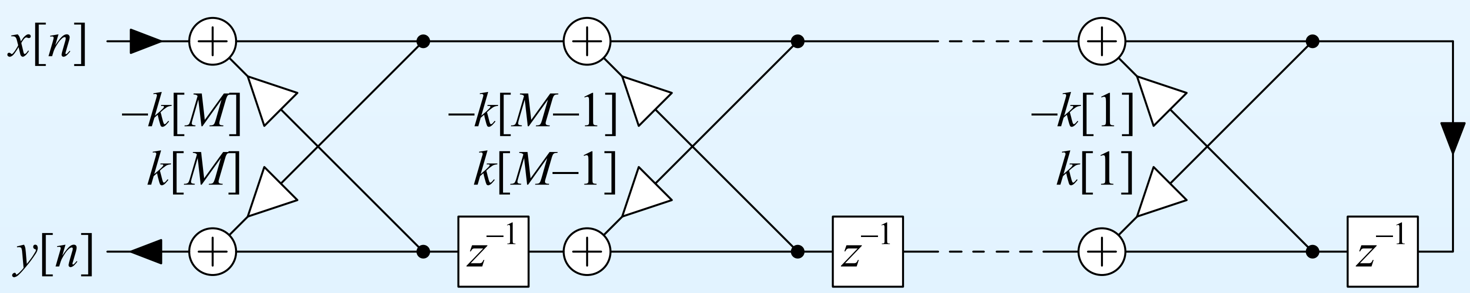


Allpass Lattice

We can implement any allpass filter $H(z) = \frac{z^{-M}A(z^{-1})}{A(z)}$ as a lattice filter with M stages:

- Initialize $A_M(z) = A(z)$
- Repeat for $m = M : -1 : 1$
  - $k[m] = a_m[m]$
  - $a_{m-1}[n] = \frac{a_m[n]-k[m]a_m[m-n]}{1-k^2[m]}$ for $0 \le n \le m-1$

    equivalently $A_{m-1}(z) = \frac{A_m(z)-k[m]z^{-m}A_m(z^{-1})}{1-k^2[m]}$



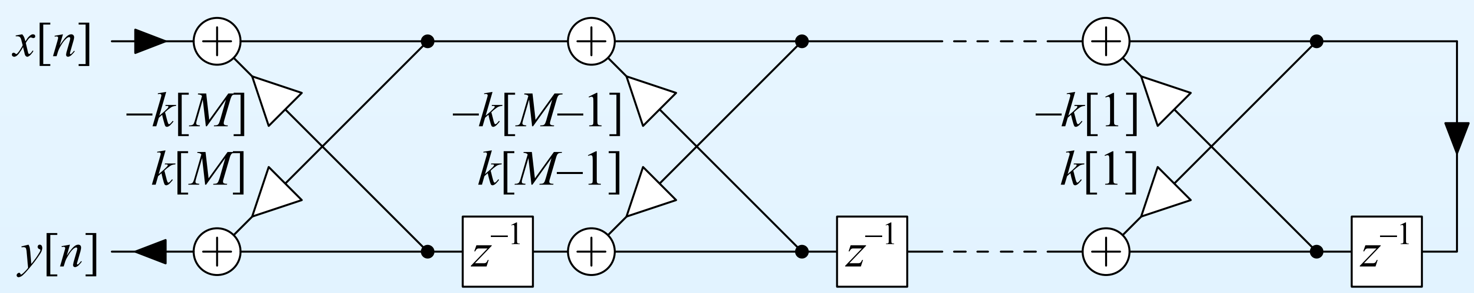
Allpass Lattice

We can implement any allpass filter $H(z) = \frac{z^{-M}A(z^{-1})}{A(z)}$ as a lattice filter with M stages:

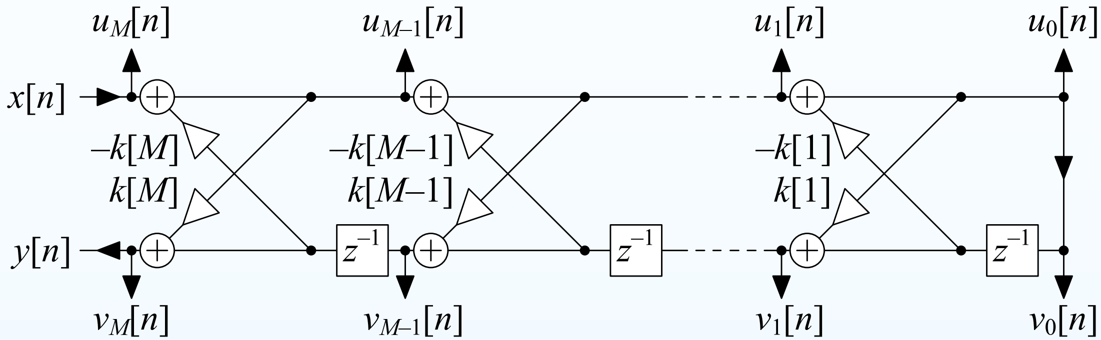
- Initialize $A_M(z) = A(z)$
- Repeat for $m = M : -1 : 1$
  - $k[m] = a_m[m]$
  - $a_{m-1}[n] = \frac{a_m[n] - k[m]a_m[m-n]}{1-k^2[m]}$ for $0 \le n \le m-1$

    equivalently $A_{m-1}(z) = \frac{A_m(z) - k[m]z^{-m}A_m(z^{-1})}{1-k^2[m]}$

$A(z)$ is stable iff $|k[m]| < 1$ for all m (good stability test)

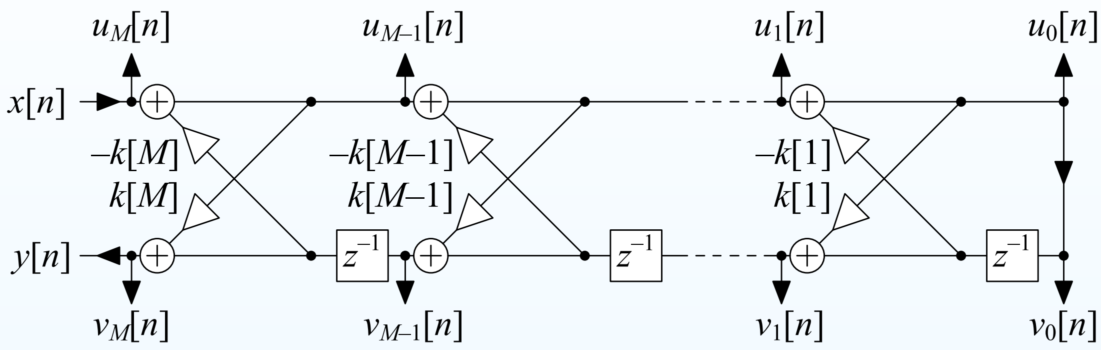


Lattice Filter



Label outputs $u_m[n]$ and $v_m[n]$ and define $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$

Lattice Filter

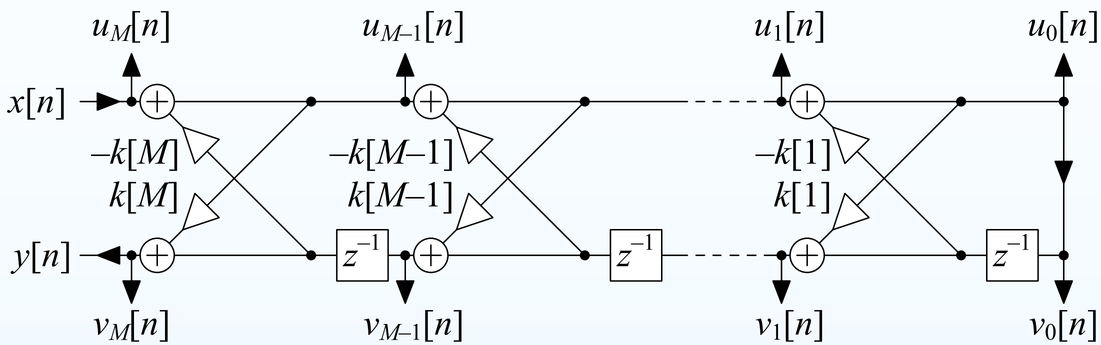


Label outputs $u_m[n]$ and $v_m[n]$ and define $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$

From earlier slide (slide 12):

$$\frac{U_{m-1}(z)}{U_m(z)} = \frac{1}{1+k[m]z^{-1}H_{m-1}(z)}$$

Lattice Filter

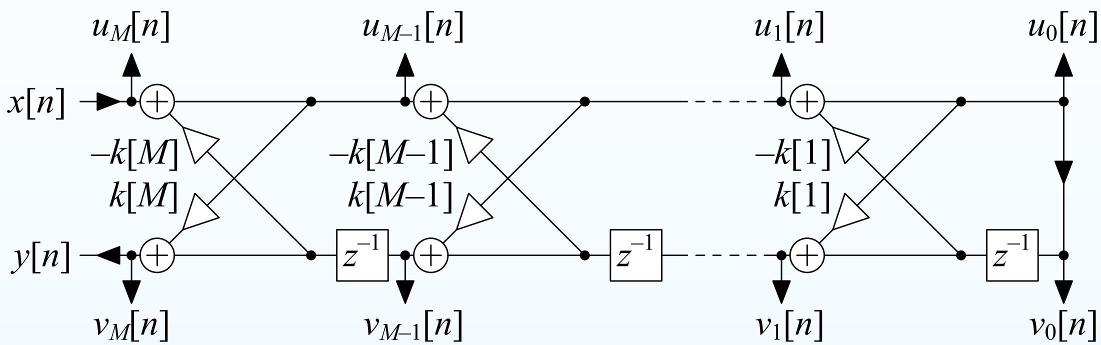


Label outputs $u_m[n]$ and $v_m[n]$ and define $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$

From earlier slide (slide 12):

$$\frac{U_{m-1}(z)}{U_m(z)} = \frac{1}{1+k[m]z^{-1}H_{m-1}(z)} = \frac{A_{m-1}(z)}{A_{m-1}(z)+k[m]z^{-m}A_{m-1}(z^{-1})}$$

Lattice Filter

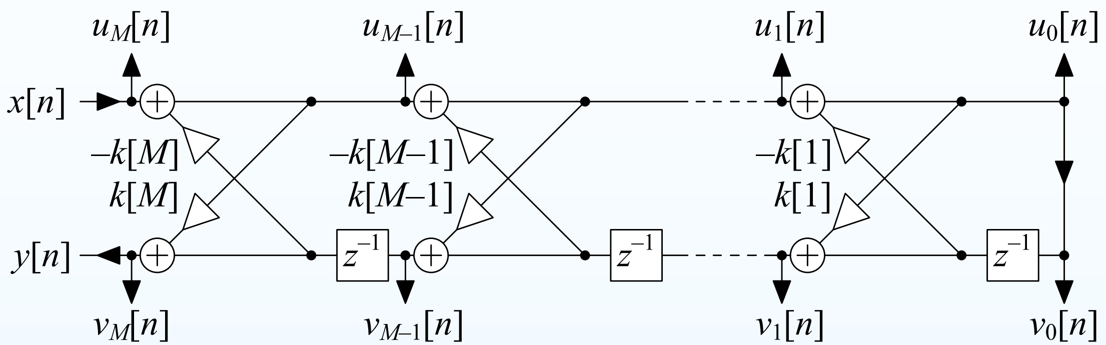


Label outputs $u_m[n]$ and $v_m[n]$ and define $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$

From earlier slide (slide 12):

$$\frac{U_{m-1}(z)}{U_m(z)} = \frac{1}{1+k[m]z^{-1}H_{m-1}(z)} = \frac{A_{m-1}(z)}{A_{m-1}(z)+k[m]z^{-m}A_{m-1}(z^{-1})} = \frac{A_{m-1}(z)}{A_m(z)}$$

Lattice Filter



Label outputs $u_m[n]$ and $v_m[n]$ and define $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$

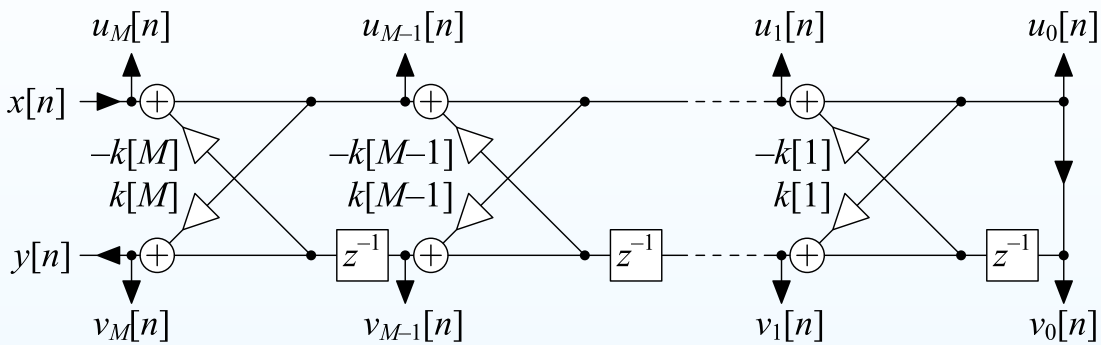
From earlier slide (slide 12):

$$\frac{U_{m-1}(z)}{U_m(z)} = \frac{1}{1+k[m]z^{-1}H_{m-1}(z)} = \frac{A_{m-1}(z)}{A_{m-1}(z)+k[m]z^{-m}A_{m-1}(z^{-1})} = \frac{A_{m-1}(z)}{A_m(z)}$$

Hence:

$$\frac{U_m(z)}{X(z)} = \frac{A_m(z)}{A(z)}$$

Lattice Filter



Label outputs $u_m[n]$ and $v_m[n]$ and define $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$

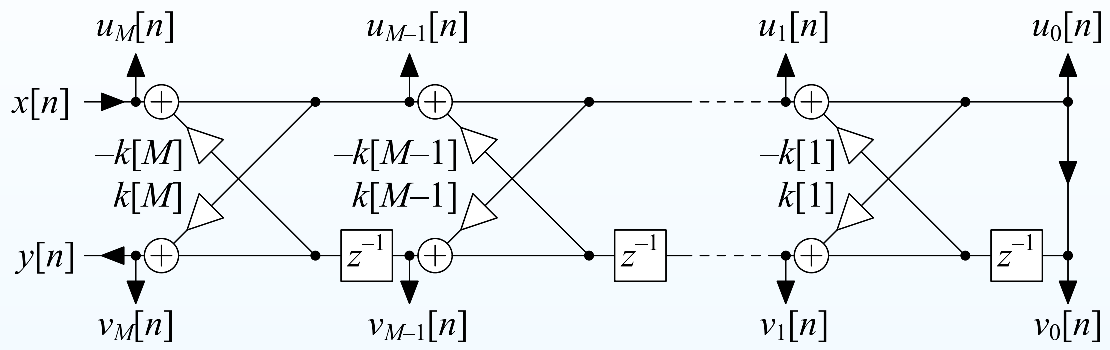
From earlier slide (slide 12):

$$\frac{U_{m-1}(z)}{U_m(z)} = \frac{1}{1+k[m]z^{-1}H_{m-1}(z)} = \frac{A_{m-1}(z)}{A_{m-1}(z)+k[m]z^{-m}A_{m-1}(z^{-1})} = \frac{A_{m-1}(z)}{A_m(z)}$$

Hence:

$$\frac{U_m(z)}{X(z)} = \frac{A_m(z)}{A(z)} \quad \text{and} \quad \frac{V_m(z)}{X(z)} = \frac{U_m(z)}{X(z)} \times \frac{V_m(z)}{U_m(z)}$$

Lattice Filter



Label outputs $u_m[n]$ and $v_m[n]$ and define $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$

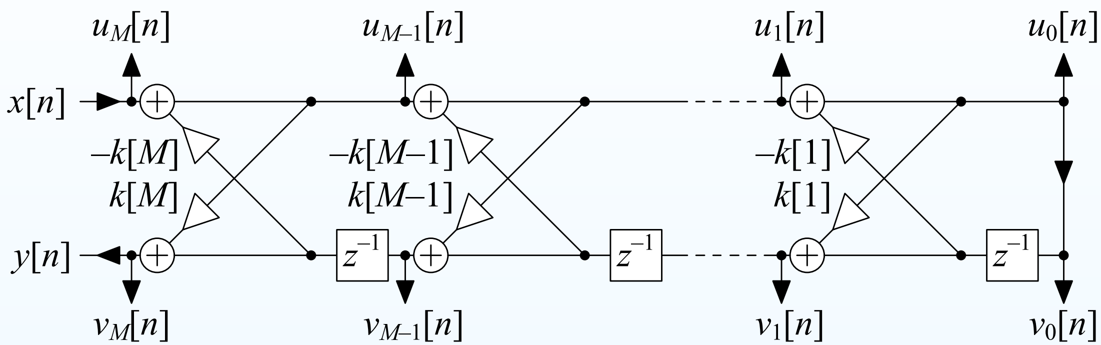
From earlier slide (slide 12):

$$\frac{U_{m-1}(z)}{U_m(z)} = \frac{1}{1+k[m]z^{-1}H_{m-1}(z)} = \frac{A_{m-1}(z)}{A_{m-1}(z)+k[m]z^{-m}A_{m-1}(z^{-1})} = \frac{A_{m-1}(z)}{A_m(z)}$$

Hence:

$$\frac{U_m(z)}{X(z)} = \frac{A_m(z)}{A(z)} \qquad \text{and} \qquad \frac{V_m(z)}{X(z)} = \frac{U_m(z)}{X(z)} \times \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A(z)}$$

Lattice Filter



Label outputs $u_m[n]$ and $v_m[n]$ and define $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$

From earlier slide (slide 12):

$$\frac{U_{m-1}(z)}{U_m(z)} = \frac{1}{1+k[m]z^{-1}H_{m-1}(z)} = \frac{A_{m-1}(z)}{A_{m-1}(z)+k[m]z^{-m}A_{m-1}(z^{-1})} = \frac{A_{m-1}(z)}{A_m(z)}$$

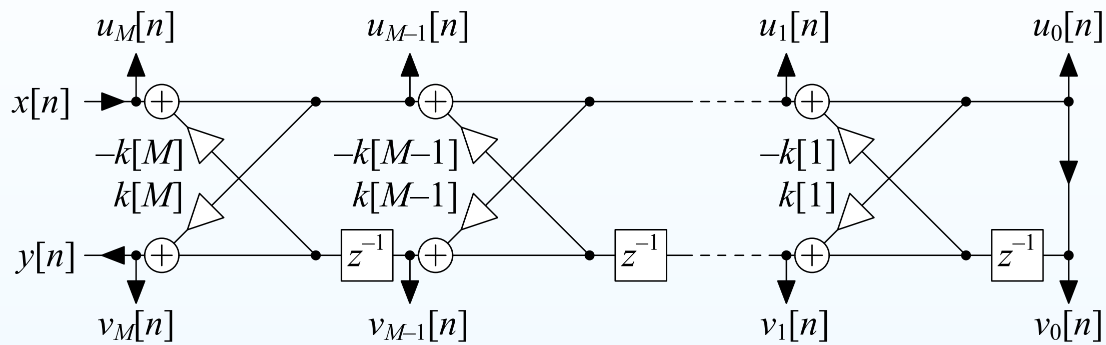
Hence:

$$\frac{U_m(z)}{X(z)} = \frac{A_m(z)}{A(z)} \qquad \text{and} \qquad \frac{V_m(z)}{X(z)} = \frac{U_m(z)}{X(z)} \times \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A(z)}$$

The numerator of $\frac{V_m(z)}{X(z)}$ is of order m so you can create any numerator of order M by summing appropriate multiples of $V_m(z)$:

$$w[n] = \sum_{m=0}^{M} c_m v_m[n]$$

Lattice Filter



Label outputs $u_m[n]$ and $v_m[n]$ and define $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$

From earlier slide (slide 12):

$$\frac{U_{m-1}(z)}{U_m(z)} = \frac{1}{1+k[m]z^{-1}H_{m-1}(z)} = \frac{A_{m-1}(z)}{A_{m-1}(z)+k[m]z^{-m}A_{m-1}(z^{-1})} = \frac{A_{m-1}(z)}{A_m(z)}$$

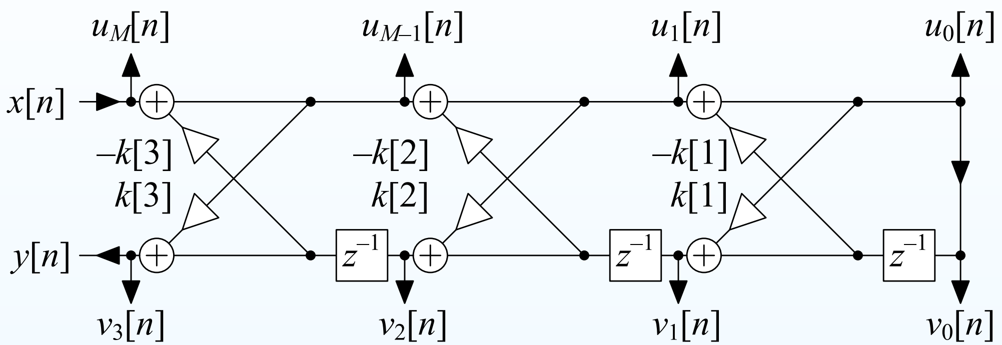
Hence:

$$\frac{U_m(z)}{X(z)} = \frac{A_m(z)}{A(z)} \qquad \text{and} \qquad \frac{V_m(z)}{X(z)} = \frac{U_m(z)}{X(z)} \times \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A(z)}$$

The numerator of $\frac{V_m(z)}{X(z)}$ is of order m so you can create any numerator of order M by summing appropriate multiples of $V_m(z)$:

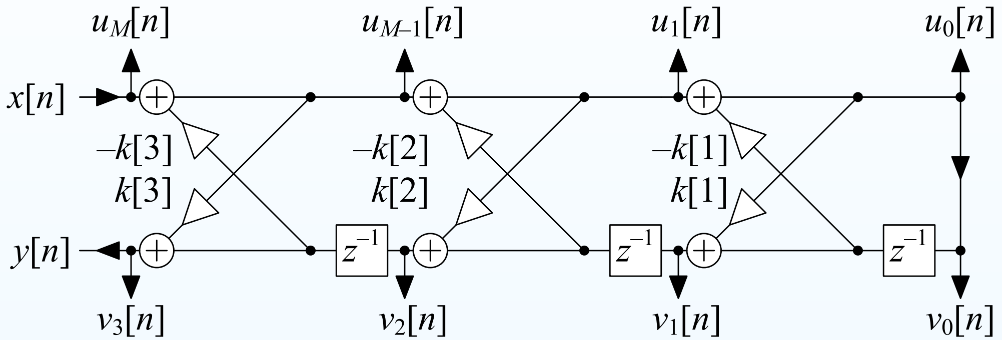
$$w[n] = \sum_{m=0}^{M} c_m v_m[n] \quad \Rightarrow \quad W(z) = \frac{\sum_{m=0}^{M} c_m z^{-m} A_m(z^{-1})}{A(z)}$$

Lattice Example



$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

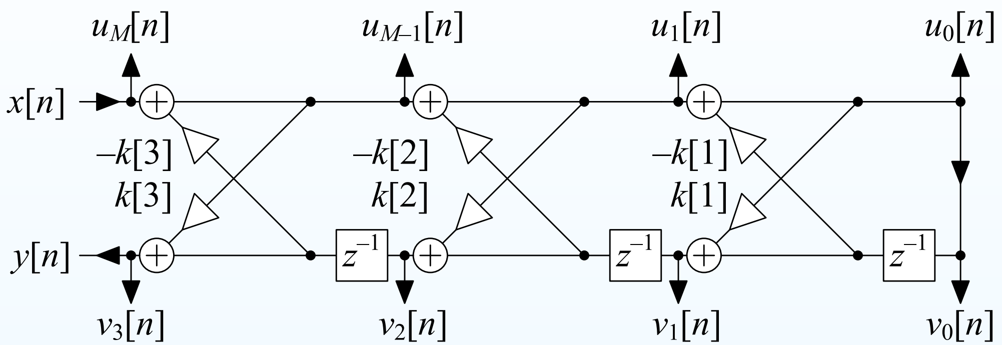
Lattice Example



$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

- $k[3] = 0.2 \Rightarrow a_2[\,] = \frac{[1,\,0.2,\,-0.23]-0.2[0.2,\,-0.23,\,0.2]}{1-0.2^2} = [1,\ 0.256,\ -0.281]$

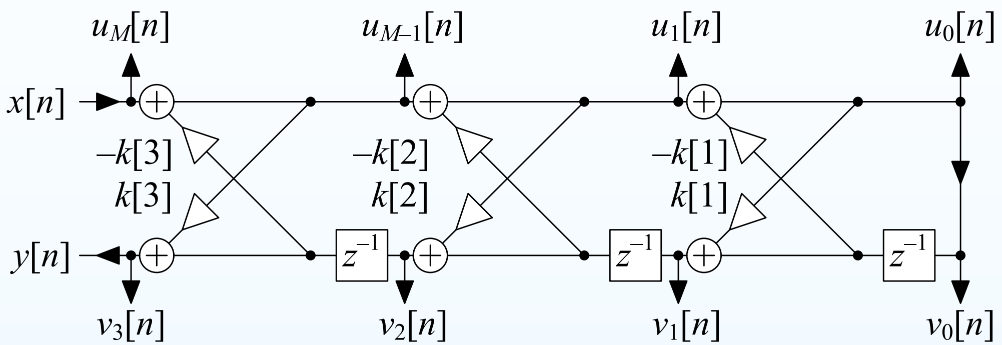
Lattice Example



$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

- $k[3] = 0.2 \Rightarrow a_2[\,] = \frac{[1,\,0.2,\,-0.23]-0.2[0.2,\,-0.23,\,0.2]}{1-0.2^2} = [1,\ 0.256,\ -0.281]$
- $k[2] = -0.281 \Rightarrow a_1[\,] = \frac{[1,\,0.256]+0.281[-0.281,\,0.256]}{1-0.281^2} = [1,\ 0.357]$

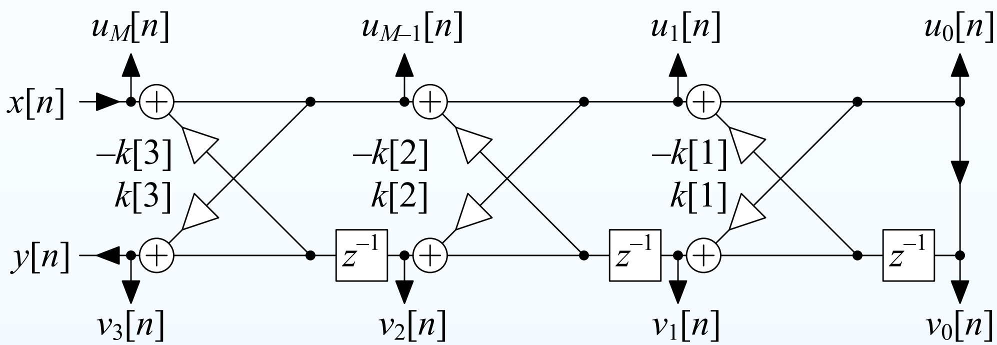
Lattice Example



$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

- $k[3] = 0.2 \Rightarrow a_2[\,] = \frac{[1,\,0.2,\,-0.23]-0.2[0.2,\,-0.23,\,0.2]}{1-0.2^2} = [1,\ 0.256,\ -0.281]$
- $k[2] = -0.281 \Rightarrow a_1[\,] = \frac{[1,\,0.256]+0.281[-0.281,\,0.256]}{1-0.281^2} = [1,\ 0.357]$
- $k[1] = 0.357 \Rightarrow a_0[\,] = 1$

Lattice Example

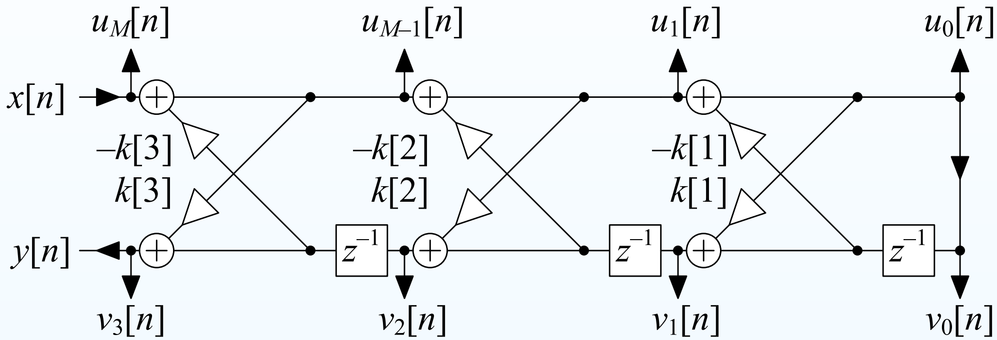


$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

- $k[3] = 0.2 \Rightarrow a_2[\,] = \frac{[1,\,0.2,\,-0.23]-0.2[0.2,\,-0.23,\,0.2]}{1-0.2^2} = [1,\ 0.256,\ -0.281]$
- $k[2] = -0.281 \Rightarrow a_1[\,] = \frac{[1,\,0.256]+0.281[-0.281,\,0.256]}{1-0.281^2} = [1,\ 0.357]$
- $k[1] = 0.357 \Rightarrow a_0[\,] = 1$

$$\frac{V_0(z)}{X(z)} = \frac{1}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

Lattice Example

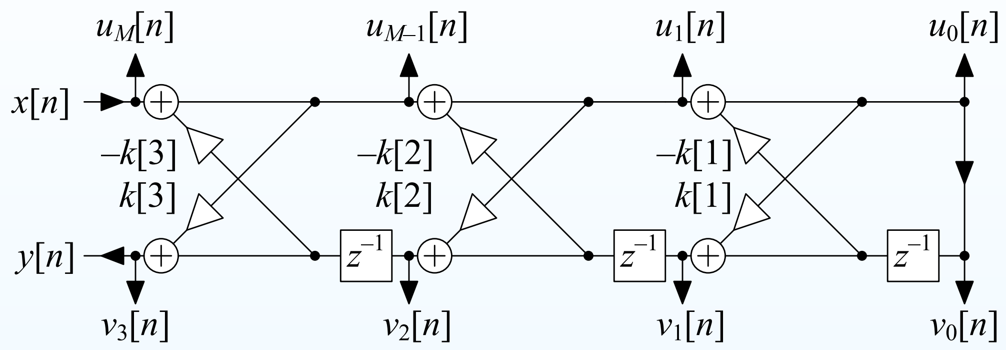


$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

- $k[3] = 0.2 \Rightarrow a_2[\,] = \frac{[1,\,0.2,\,-0.23]-0.2[0.2,\,-0.23,\,0.2]}{1-0.2^2} = [1,\ 0.256,\ -0.281]$
- $k[2] = -0.281 \Rightarrow a_1[\,] = \frac{[1,\,0.256]+0.281[-0.281,\,0.256]}{1-0.281^2} = [1,\ 0.357]$
- $k[1] = 0.357 \Rightarrow a_0[\,] = 1$

$$\frac{V_0(z)}{X(z)} = \frac{1}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_1(z)}{X(z)} = \frac{0.357+z^{-1}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

Lattice Example



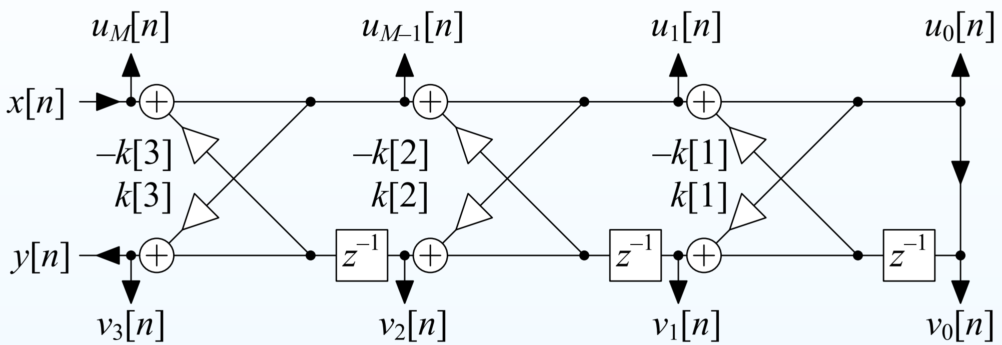
$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

- $k[3] = 0.2 \Rightarrow a_2[\,] = \frac{[1,\ 0.2,\ -0.23] - 0.2[0.2,\ -0.23,\ 0.2]}{1-0.2^2} = [1,\ 0.256,\ -0.281]$
- $k[2] = -0.281 \Rightarrow a_1[\,] = \frac{[1,\ 0.256] + 0.281[-0.281,\ 0.256]}{1-0.281^2} = [1,\ 0.357]$
- $k[1] = 0.357 \Rightarrow a_0[\,] = 1$

$$\frac{V_0(z)}{X(z)} = \frac{1}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_1(z)}{X(z)} = \frac{0.357+z^{-1}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

$$\frac{V_2(z)}{X(z)} = \frac{-0.281+0.256z^{-1}+z^{-2}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

Lattice Example



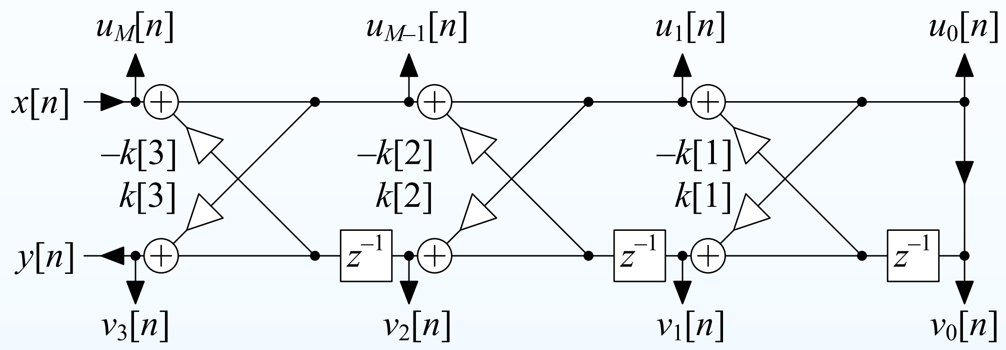
$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

- $k[3] = 0.2 \Rightarrow a_2[\,] = \frac{[1,\ 0.2,\ -0.23] - 0.2[0.2,\ -0.23,\ 0.2]}{1-0.2^2} = [1,\ 0.256,\ -0.281]$
- $k[2] = -0.281 \Rightarrow a_1[\,] = \frac{[1,\ 0.256] + 0.281[-0.281,\ 0.256]}{1-0.281^2} = [1,\ 0.357]$
- $k[1] = 0.357 \Rightarrow a_0[\,] = 1$

$$\frac{V_0(z)}{X(z)} = \frac{1}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_1(z)}{X(z)} = \frac{0.357+z^{-1}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

$$\frac{V_2(z)}{X(z)} = \frac{-0.281+0.256z^{-1}+z^{-2}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_3(z)}{X(z)} = \frac{0.2-0.23z^{-1}+0.2z^{-2}+z^{-3}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

Lattice Example



$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

- $k[3] = 0.2 \Rightarrow a_2[\,] = \frac{[1,\,0.2,\,-0.23]-0.2[0.2,\,-0.23,\,0.2]}{1-0.2^2} = [1,\ 0.256,\ -0.281]$
- $k[2] = -0.281 \Rightarrow a_1[\,] = \frac{[1,\,0.256]+0.281[-0.281,\,0.256]}{1-0.281^2} = [1,\ 0.357]$
- $k[1] = 0.357 \Rightarrow a_0[\,] = 1$

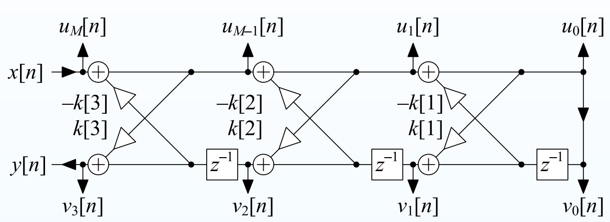
$$\frac{V_0(z)}{X(z)} = \frac{1}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_1(z)}{X(z)} = \frac{0.357+z^{-1}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

$$\frac{V_2(z)}{X(z)} = \frac{-0.281+0.256z^{-1}+z^{-2}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_3(z)}{X(z)} = \frac{0.2-0.23z^{-1}+0.2z^{-2}+z^{-3}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

Add together multiples of $\frac{V_m(z)}{X(z)}$ to create an arbitrary $\frac{B(z)}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$

Lattice Example Numerator

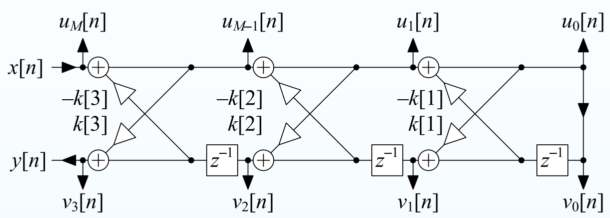
Form a new output signal as $w[n] = \sum_{m=0}^{M} c_m v_m[n]$



Lattice Example Numerator

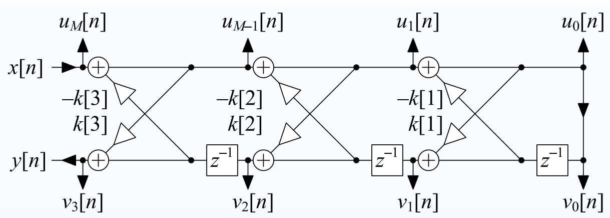
Form a new output signal as $w[n] = \sum_{m=0}^{M} c_m v_m[n]$

$W(z) = \sum_{m=0}^{M} c_m V_m(z) = \frac{B(z)}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} X(z)$



Lattice Example Numerator

Form a new output signal as $w[n] = \sum_{m=0}^{M} c_m v_m[n]$



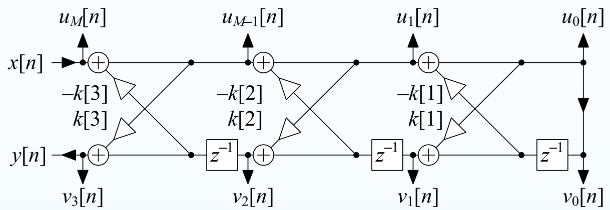
$W(z) = \sum_{m=0}^{M} c_m V_m(z) = \frac{B(z)}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} X(z)$

$$\frac{V_0(z)}{X(z)} = \frac{1}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_1(z)}{X(z)} = \frac{0.357+z^{-1}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

$$\frac{V_2(z)}{X(z)} = \frac{-0.281+0.256z^{-1}+z^{-2}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_3(z)}{X(z)} = \frac{0.2-0.23z^{-1}+0.2z^{-2}+z^{-3}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

Lattice Example Numerator

Form a new output signal as $w[n] = \sum_{m=0}^{M} c_m v_m[n]$



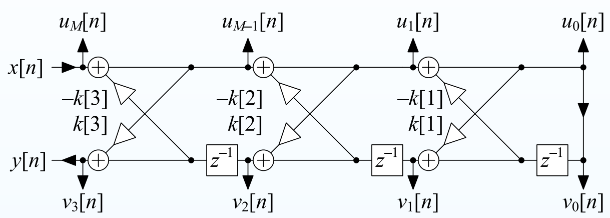
$W(z) = \sum_{m=0}^{M} c_m V_m(z) = \frac{B(z)}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} X(z)$

$$\frac{V_0(z)}{X(z)} = \frac{1}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_1(z)}{X(z)} = \frac{0.357+z^{-1}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

$$\frac{V_2(z)}{X(z)} = \frac{-0.281+0.256z^{-1}+z^{-2}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_3(z)}{X(z)} = \frac{0.2-0.23z^{-1}+0.2z^{-2}+z^{-3}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

We have $\begin{pmatrix} b[0] \\ b[1] \\ b[2] \\ b[3] \end{pmatrix} = \begin{pmatrix} 1 & 0.357 & -0.281 & 0.2 \\ 0 & 1 & 0.256 & -0.23 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$

Lattice Example Numerator



Form a new output signal as $w[n] = \sum_{m=0}^{M} c_m v_m[n]$

$W(z) = \sum_{m=0}^{M} c_m V_m(z) = \frac{B(z)}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} X(z)$

$$\frac{V_0(z)}{X(z)} = \frac{1}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_1(z)}{X(z)} = \frac{0.357+z^{-1}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

$$\frac{V_2(z)}{X(z)} = \frac{-0.281+0.256z^{-1}+z^{-2}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}} \qquad \frac{V_3(z)}{X(z)} = \frac{0.2-0.23z^{-1}+0.2z^{-2}+z^{-3}}{1+0.2z^{-1}-0.23z^{-2}+0.2z^{-3}}$$

We have $\begin{pmatrix} b[0] \\ b[1] \\ b[2] \\ b[3] \end{pmatrix} = \begin{pmatrix} 1 & 0.357 & -0.281 & 0.2 \\ 0 & 1 & 0.256 & -0.23 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$

Hence choose c_m as $\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 & 0.357 & -0.281 & 0.2 \\ 0 & 1 & 0.256 & -0.23 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} b[0] \\ b[1] \\ b[2] \\ b[3] \end{pmatrix}$

Summary

- Filter block diagrams
  - Direct forms
  - Transposition
  - State space representation

Summary

- Filter block diagrams
  - Direct forms
  - Transposition
  - State space representation
- Precision issues: coefficient error, arithmetic error
  - cascaded biquads

Summary

- Filter block diagrams
  - Direct forms
  - Transposition
  - State space representation
- Precision issues: coefficient error, arithmetic error
  - cascaded biquads
- Allpass filters
  - first and second order sections

Summary

- Filter block diagrams
  - Direct forms
  - Transposition
  - State space representation
- Precision issues: coefficient error, arithmetic error
  - cascaded biquads
- Allpass filters
  - first and second order sections
- Lattice filters
  - Arbitrary allpass response
  - Arbitrary IIR response by summing intermediate outputs

Summary

- Filter block diagrams
  - Direct forms
  - Transposition
  - State space representation
- Precision issues: coefficient error, arithmetic error
  - cascaded biquads
- Allpass filters
  - first and second order sections
- Lattice filters
  - Arbitrary allpass response
  - Arbitrary IIR response by summing intermediate outputs

For further details see Mitra: 8.

MATLAB routines

residuez	$\frac{b(z^{-1})}{a(z^{-1})} \rightarrow \sum_k \frac{r_k}{1-p_k z^{-1}}$
tf2sos,sos2tf	$\frac{b(z^{-1})}{a(z^{-1})} \leftrightarrow \prod_l \frac{b_{0,l}+b_{1,l}z^{-1}+b_{2,l}z^{-2}}{1+a_{1,l}z^{-1}+a_{2,l}z^{-2}}$
zp2sos,sos2zp	$\{z_m, p_k, g\} \leftrightarrow \prod_l \frac{b_{0,l}+b_{1,l}z^{-1}+b_{2,l}z^{-2}}{1+a_{\in 1,l}z^{-1}+a_{2,l}z^{-2}}$
zp2ss,ss2zp	$\{z_m, p_k, g\} \leftrightarrow \begin{cases} x' = Ax + Bu \\ y = Cx + Du \end{cases}$
tf2ss,ss2tf	$\frac{b(z^{-1})}{a(z^{-1})} \leftrightarrow \begin{cases} x' = Ax + Bu \\ y = Cx + Du \end{cases}$
poly	$\text{poly}(\mathbf{A}) = \det(z\mathbf{I}-\mathbf{A})$