EITF75 Systems and Signals

Some more DFT

$a_0 = 6 \quad a_1 = 4 \quad a_2 = 5 \quad \ldots$

What is

645275638537458374747857845576578213012365107658014
273651820375647817891457498274584365784236571248684
172394403692032305868704230214928536574812345676574
830109476802058837567382212001039376843492838692329
483285746748239293901092056008076054328762456475867
868457387489436789546738427432659843768578456776876
548257210436903291498547654541859627376578908283675

*

769823768923758241040350524048376572365746572364736
573625748563723657465723654726357463257674326547236
574633265723865743652736574832652738657843652783562
785627856237865873465723865723657286527836527865872
365827652738657326523981524002496084867542736572357
843827489237584672890310927483975653238443027902951
990240068265668920847567896524123988764536895600263

EITF75 Systems and Signals

Some more DFT

For any number $A = \sum a_k 10^k$

$$B = \sum b_k 10^k$$

$$C = \ A \cdot B = \left(\sum a_k 10^k\right)\left(\sum b_k 10^k\right) = \sum c_k 10^k$$

EITF75 Systems and Signals

Some more DFT

For any number $$A = \sum a_k 10^k$$

$$B = \sum b_k 10^k$$

$$C = \; A \cdot B = \left(\sum a_k 10^k\right)\left(\sum b_k 10^k\right) = \sum c_k 10^k$$

The sequence $\{a_k\}$ has a z-transform (it is a finite signal)

$$A(z) = \sum a_k z^{-k}$$

ROC: z>0

EITF75 Systems and Signals

Some more DFT

For any number $$A = \sum a_k 10^k$$

$$B = \sum b_k 10^k$$

$$C = \ A \cdot B = \left(\sum a_k 10^k\right)\left(\sum b_k 10^k\right) = \sum c_k 10^k$$

The sequence $\{a_k\}$ has a z-transform (it is a finite signal)

$$A(z) = \sum a_k z^{-k}$$

ROC: z>0

Thus,

$$A = \underline{A(z)}\Big|_{z=1/10}$$

$$B = B(z)\Big|_{z=1/10}$$

$$C = C(z)\Big|_{z=1/10}$$

$$C(z) = A(z)B(z)$$

$$c_k = a_k \star b_k$$

EITF75 Systems and Signals

Some more DFT

For any number $A = \sum a_k 10^k$

$$B = \sum b_k 10^k$$

To get $\{c_k\}$:
- Convolution of long signals
- Zero-pad
- Take DFTs
- Multiply
- Take IDFT

$$C = A \cdot B = \left(\sum a_k 10^k\right)\left(\sum b_k 10^k\right) = \sum c_k 10^k$$

The sequence $\{a_k\}$ has a z-transform (it is a finite signal)

$$A(z) = \sum a_k z^{-k}$$

ROC: z>0

Thus,

$$A = A(z)\big|_{z=1/10}$$

$$B = B(z)\big|_{z=1/10}$$

$$C = C(z)\big|_{z=1/10}$$

$$C(z) = A(z)B(z)$$

$$c_k = a_k \star b_k$$

EITF75 Systems and Signals

Some more DFT

For any number $A = \sum a_k 10^k$

$B = \sum b_k 10^k$

$$C = A \cdot B = \left(\sum a_k 10^k\right)\left(\sum b_k 10^k\right) = \sum c_k 10^k$$

To get $\{c_k\}$:
- Convolution of long signals
- Zero-pad
- Take DFTs
- Multiply
- Take IDFT

If some $c_k>9$:
- Subtract 10
- Add 1 to c_{k+1}

The sequence $\{a_k\}$ has a z-transform (it is a finite signal)

$$A(z) = \sum a_k z^{-k}$$

Thus,

$$A = A(z)\big|_{z=1/10}$$

$$B = B(z)\big|_{z=1/10}$$

$$C = C(z)\big|_{z=1/10}$$

To get C:
- Write down c_k without spaces

Complexity: $N\log_2(N)$

N being size of DFT, i.e., number of digits of numbers to be multiplied.

Matlab handles N=100000 EASILY

EITF75 Systems and Signals

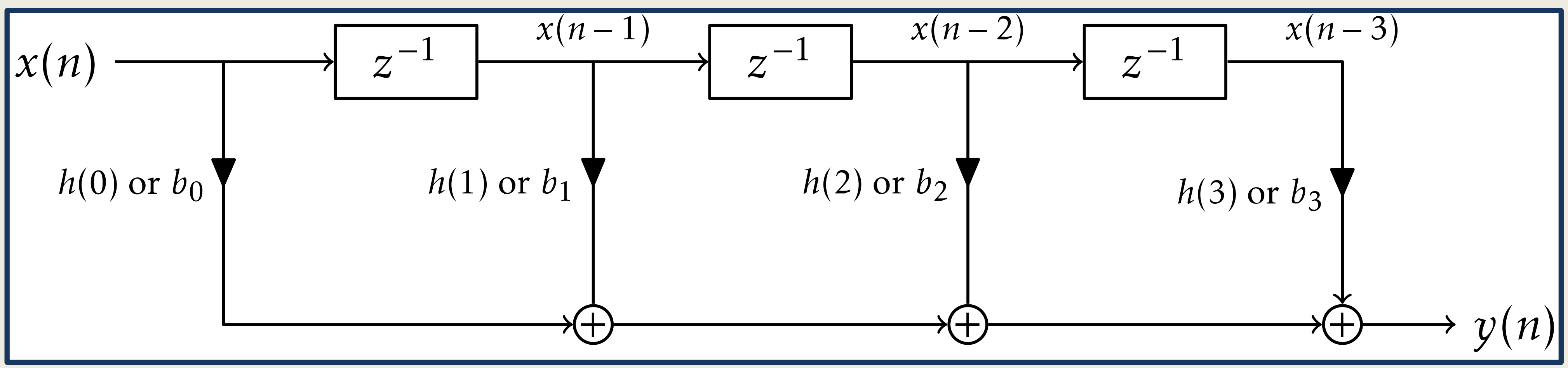
Some implementation aspects

Start with FIR filters

$$y(n) = \sum_{k=0}^{K} h(k)x(n-k) = \sum_{k=0}^{K} b_k x(n-k)$$

(notation for difference equation)

Easy to see that this is an implementation. Direct form I



Slight problem: If we need to add another zero, <u>all</u> taps change

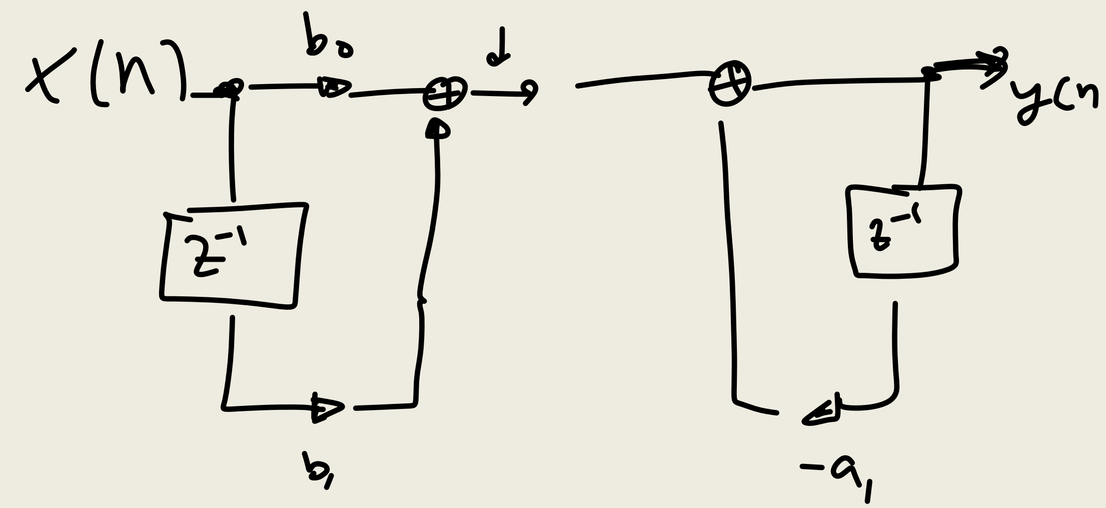
EITF75 Systems and Signals

Some implementation aspects

IIR filters (More to come on FIR filters)

$$y(n) + a_1 y(n-1) = b_0 x(n) + b_1 x(n-1)$$

$$y(n) = b_0 x(n) + b_1 x(n-1) - a_1 y(n-1)$$

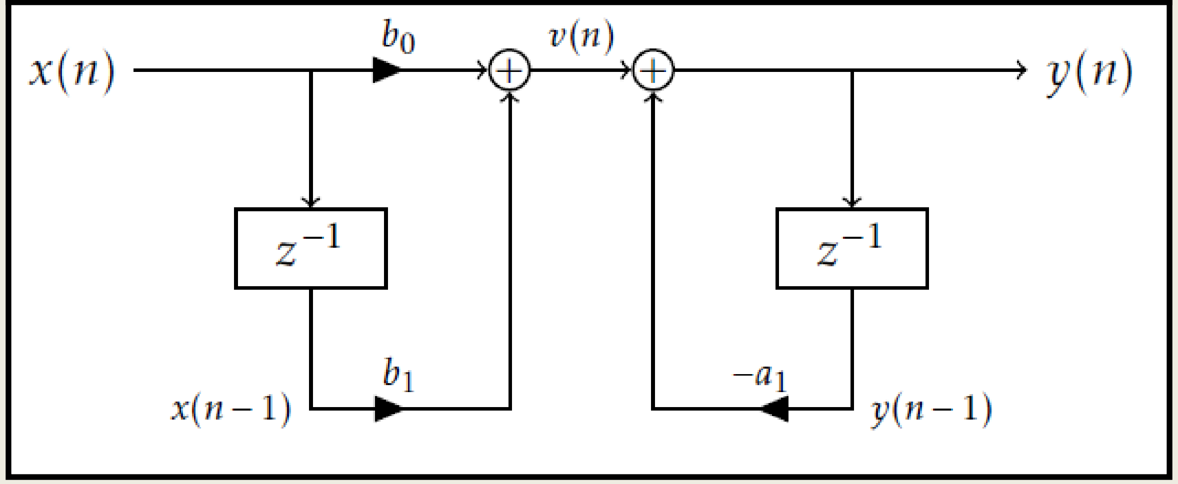


EITF75 Systems and Signals

Some implementation aspects

IR filters (More to come on FIR filters)

$$y(n) + a_1 y(n-1) = b_0 x(n) + b_1 x(n-1)$$



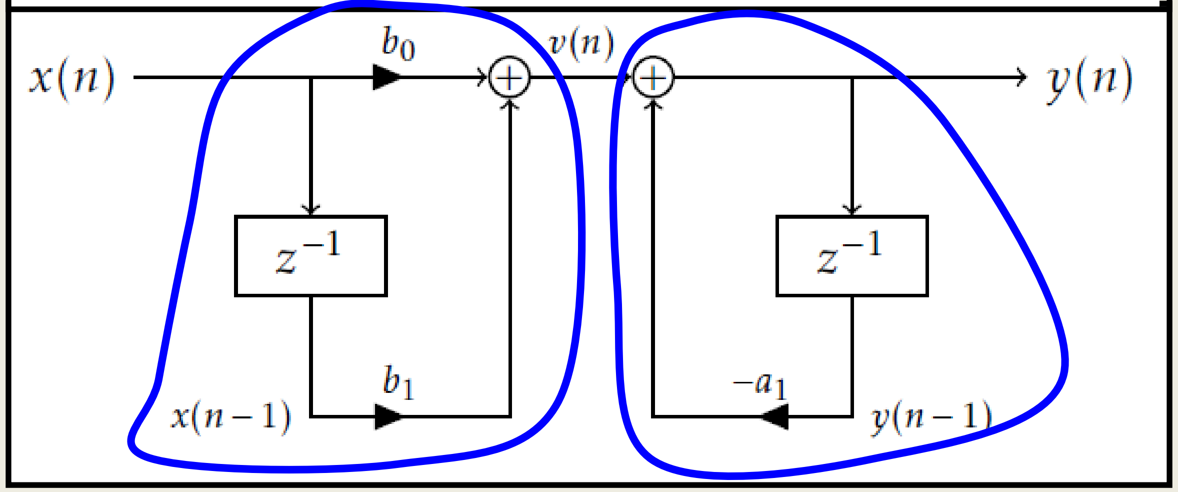
EITF75 Systems and Signals

Some implementation aspects

IIR filters

(More to come on FIR filters)

$$y(n) + a_1 y(n-1) = b_0 x(n) + b_1 x(n-1)$$



$$b_0 + b_1 z^{-1} \qquad \frac{1}{1 - a_1 z^{-1}}$$

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

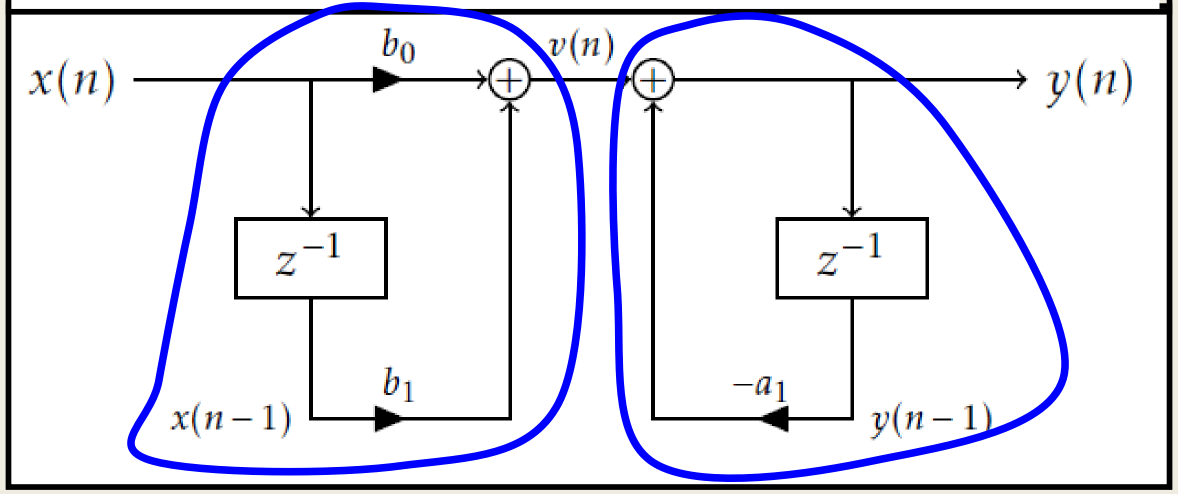
$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}}$$

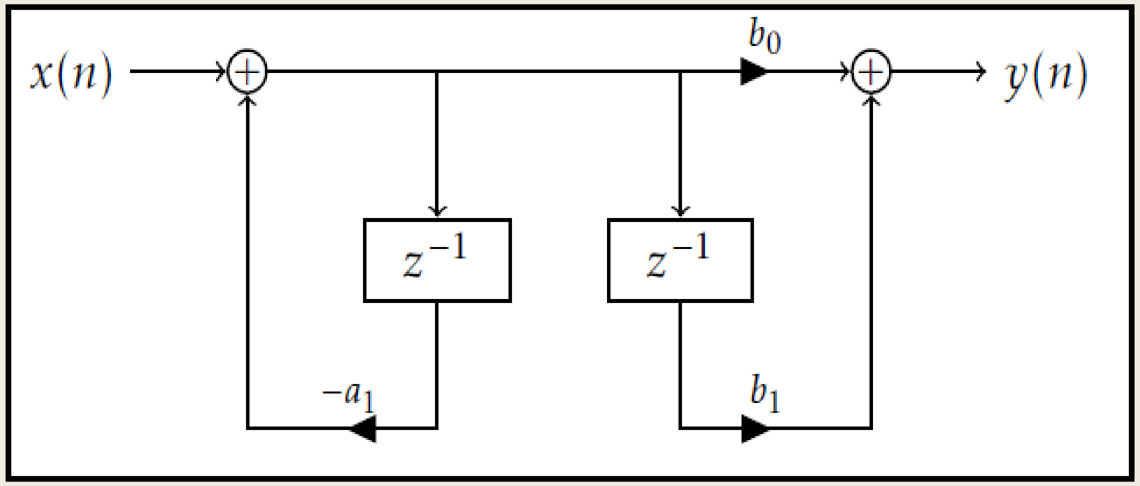
EITF75 Systems and Signals

Some implementation aspects

IIR filters (More to come on FIR filters)

$$y(n) + a_1 y(n-1) = b_0 x(n) + b_1 x(n-1)$$





$b_0 + b_1 z^{-1}$ $\qquad \dfrac{1}{1 - a_1 z^{-1}}$

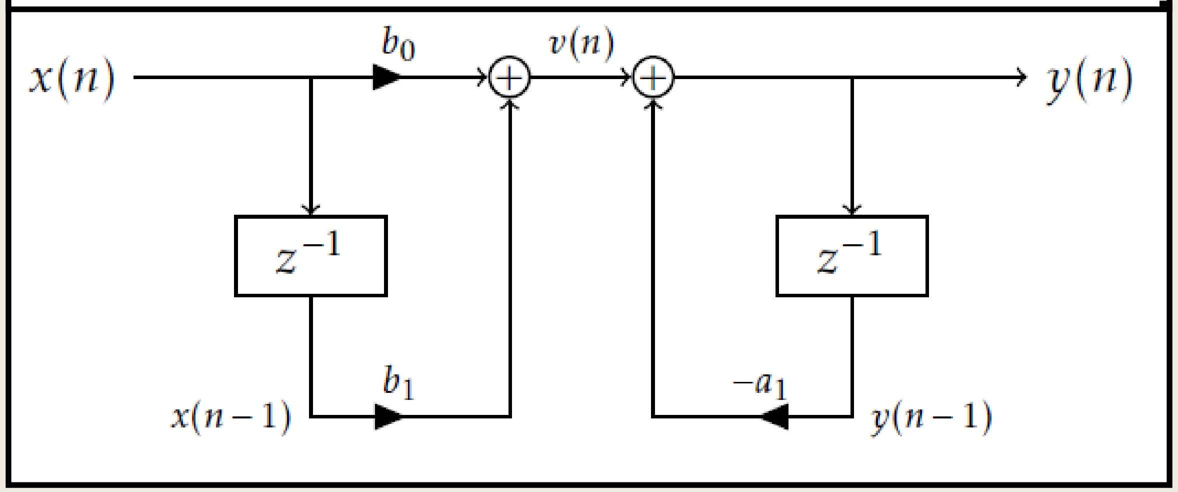
$$H(z) = (b_0 + b_1 z^{-1}) \cdot \frac{1}{1 - a_1 z^{-1}} = \frac{1}{1 - a_1 z^{-1}} \cdot (b_0 + b_1 z^{-1})$$

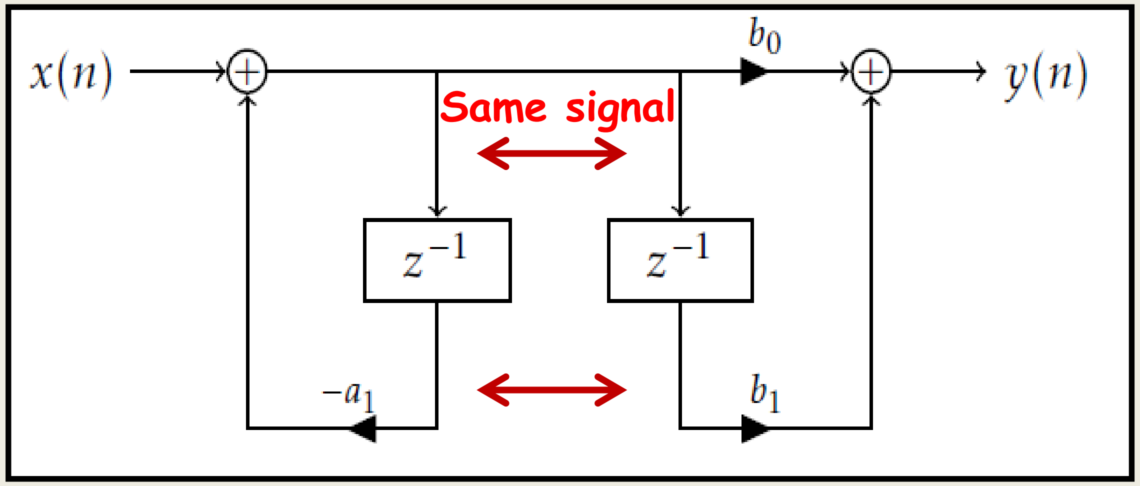
EITF75 Systems and Signals

Some implementation aspects

IIR filters (More to come on FIR filters)

$$y(n) + a_1 y(n-1) = b_0 x(n) + b_1 x(n-1)$$



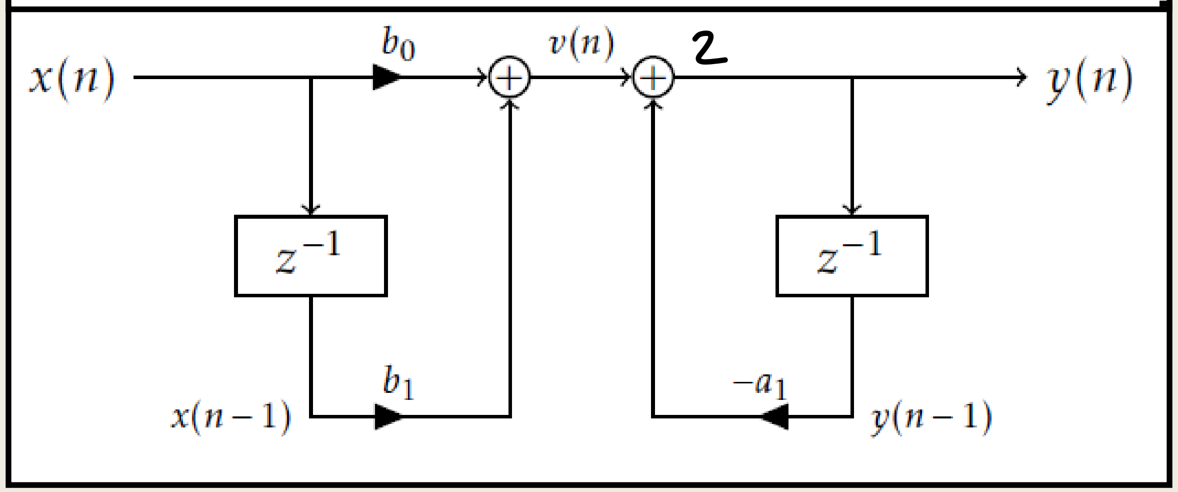


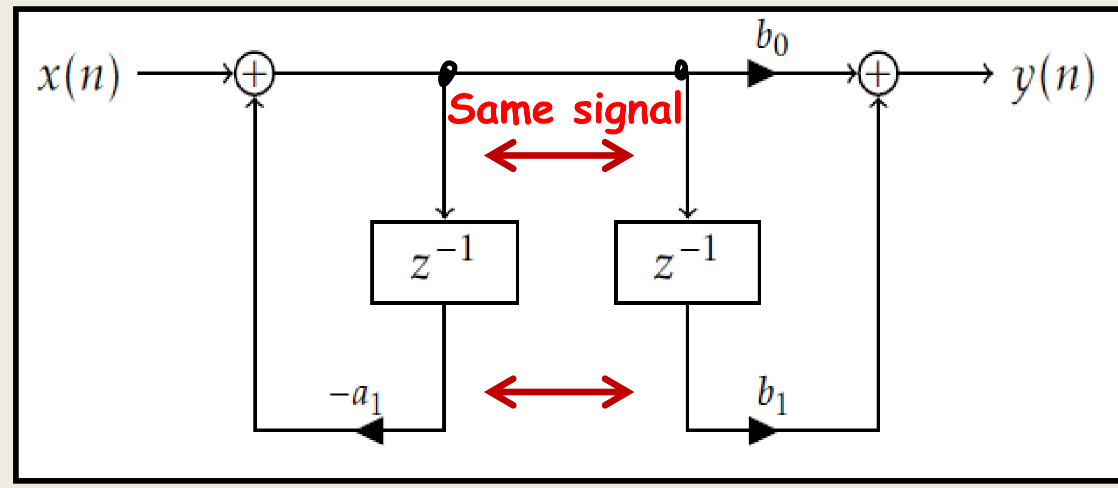
Some implementation aspects

IIR filters

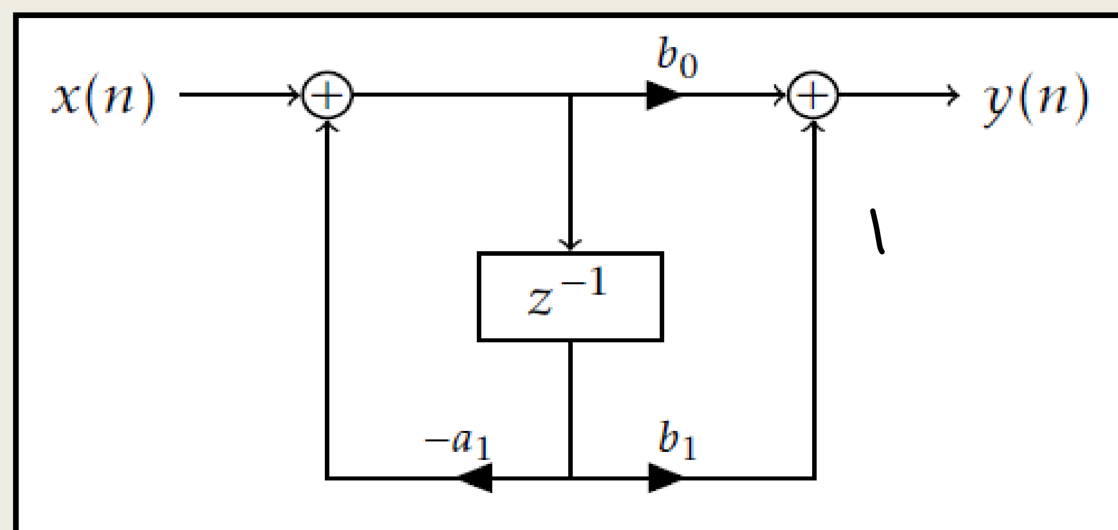
(More to come on FIR filters)

$$y(n) + a_1 y(n-1) = b_0 x(n) + b_1 x(n-1)$$





Direct form II

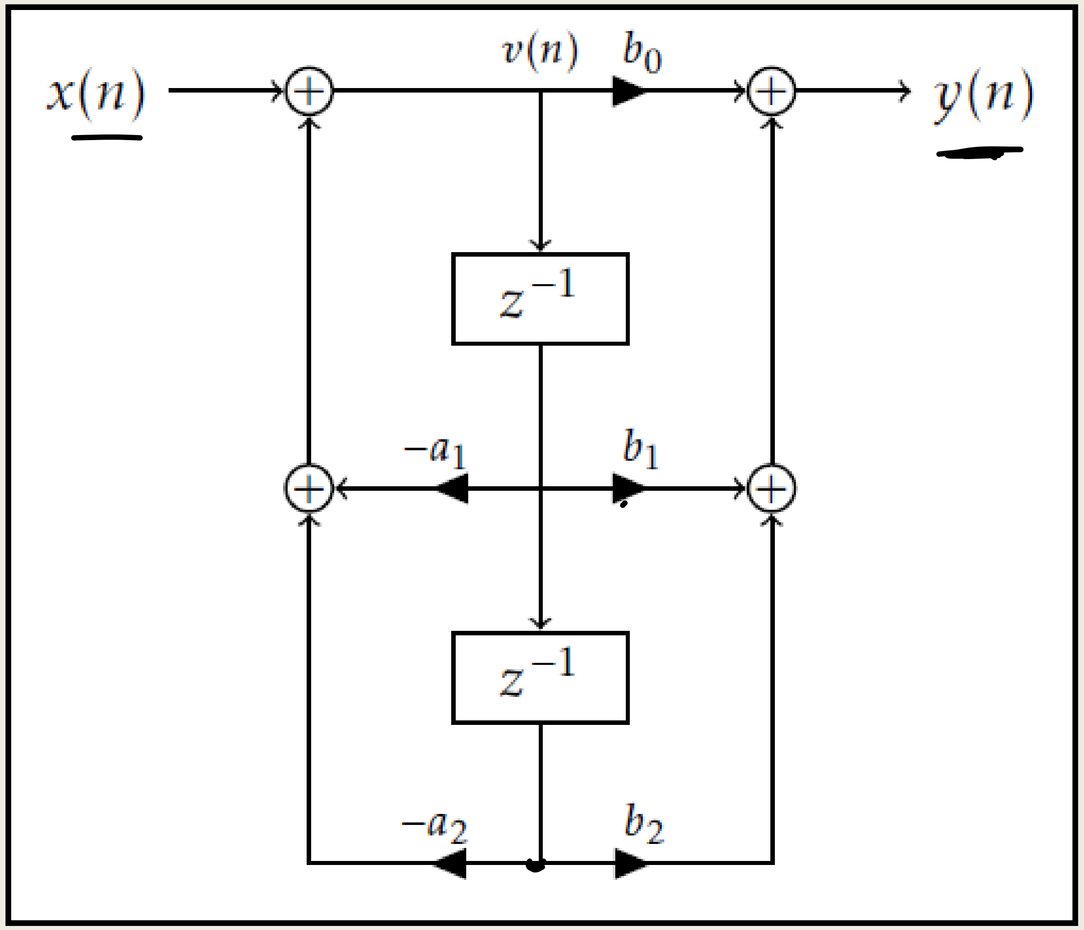


EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



$$V(z) = -z^{-1}a_1 V(z) - z^{-2}a_2 V(z) + X(z)$$

$$V(z)\left[1 + z^{-1}a_1 + z^{-2}a_2\right] = X(z)$$

$$V(z) = \frac{X(z)}{1 + z^{-1}a_1 + z^{-2}a_2}$$

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

$$Y(z) = V(z)\left[b_0 + z^{-1}b_1 + z^{-2}b_2\right]$$

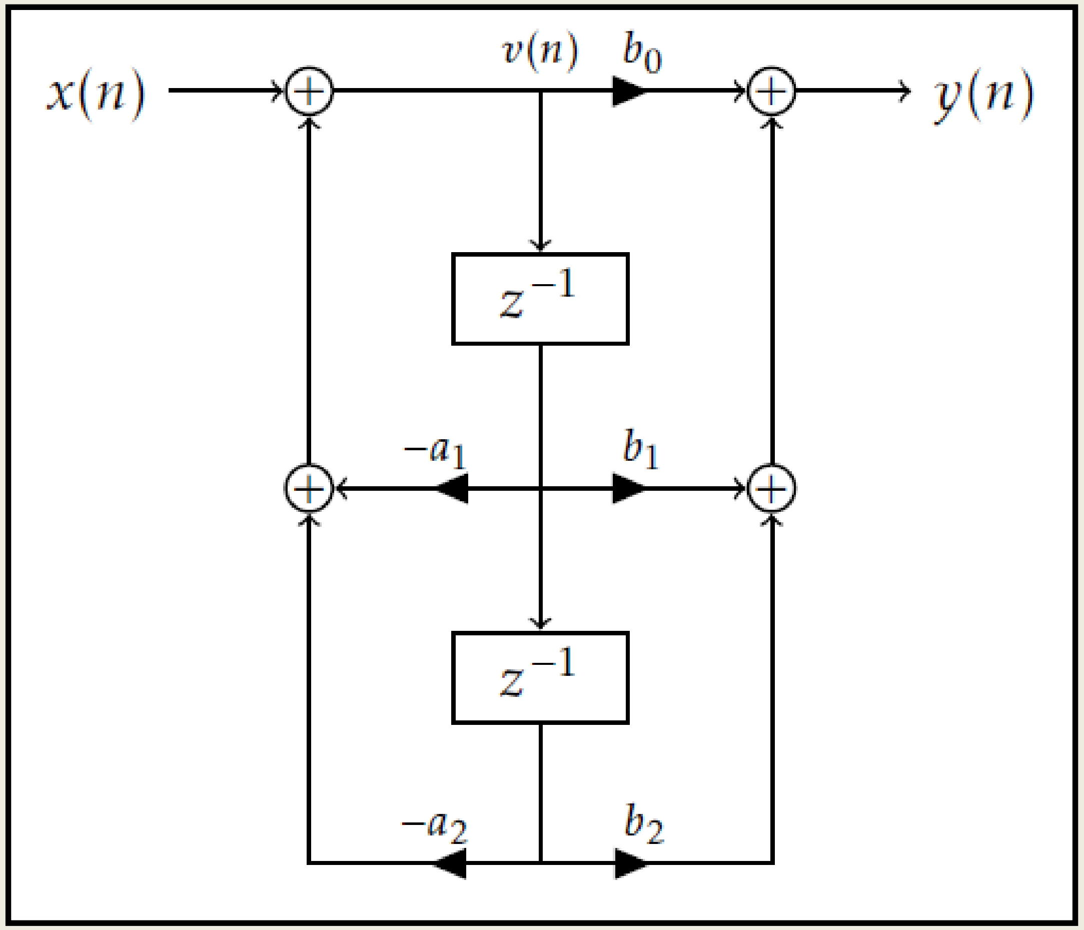
$$Y(z) = \frac{b_0 + z^{-1}b_1 + z^{-2}b_2}{1 + z^{-1}a_1 + z^{-2}a_2} X(z)$$

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



$$V(z) = -z^{-1}a_1 V(z) - z^{-2}a_2 V(z) + X(z)$$

$$V(z) + z^{-1}a_1 V(z) + z^{-2}a_2 V(z) = X(z)$$

$$V(z)\cdot\left(1 + z^{-1}a_1 + z^{-2}a_2\right) = X(z)$$

$$V(z) = \frac{X(z)}{1 + z^{-1}a_1 + z^{-2}a_2}$$ IIR part

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

$$Y(z) = V(z)\cdot\left(b_0 + z^{-1}b_1 + z^{-2}b_2\right)$$

FIR part

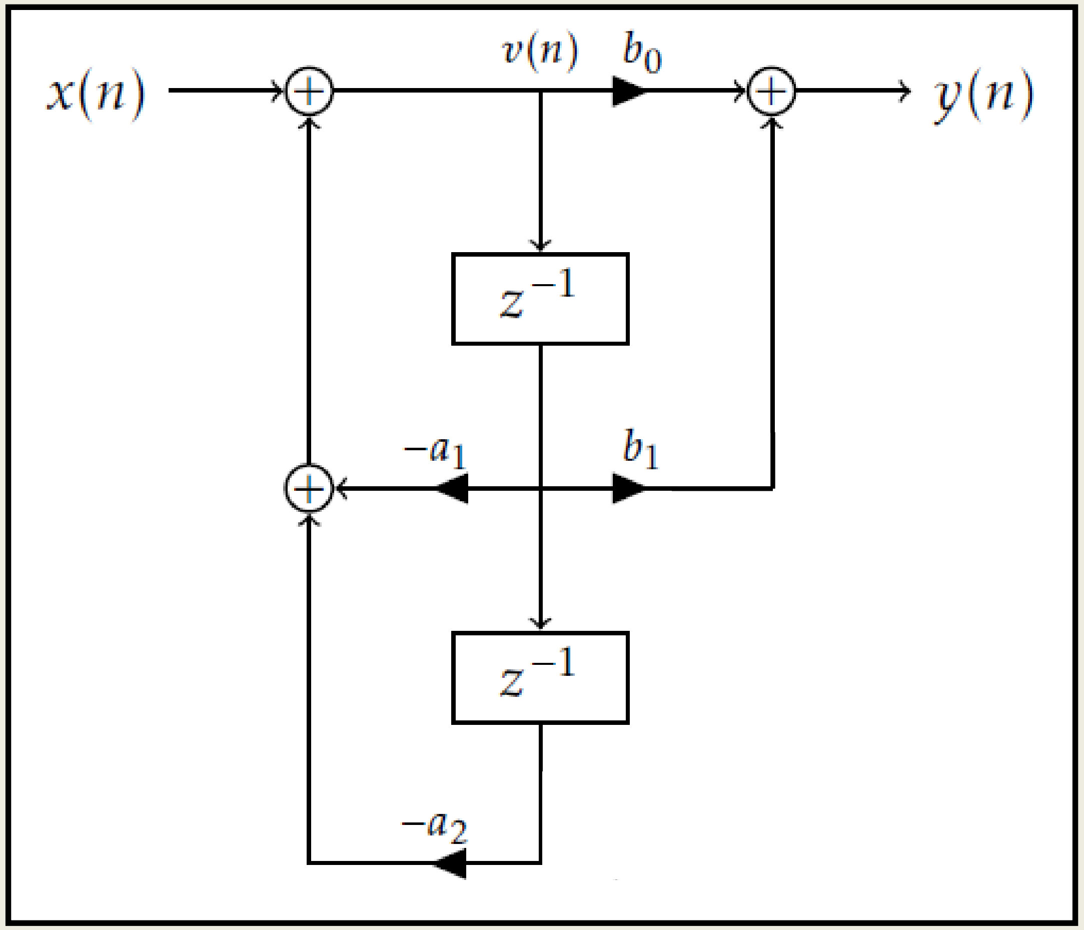
$$Y(z) = \frac{b_0 + z^{-1}b_1 + z^{-2}b_2}{1 + z^{-1}a_1 + z^{-2}a_2}\cdot X(z)$$

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



$$V(z) = -z^{-1}a_1 V(z) - z^{-2}a_2 V(z) + X(z)$$

$$V(z) + z^{-1}a_1 V(z) + z^{-2}a_2 V(z) = X(z)$$

$$V(z)\cdot\left(1 + z^{-1}a_1 + z^{-2}a_2\right) = X(z)$$

$$V(z) = \frac{X(z)}{1 + z^{-1}a_1 + z^{-2}a_2}$$ IIR part

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

$$Y(z) = V(z)\cdot\left(b_0 + z^{-1}b_1 + z^{-2}b_2\right)$$

FIR part

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

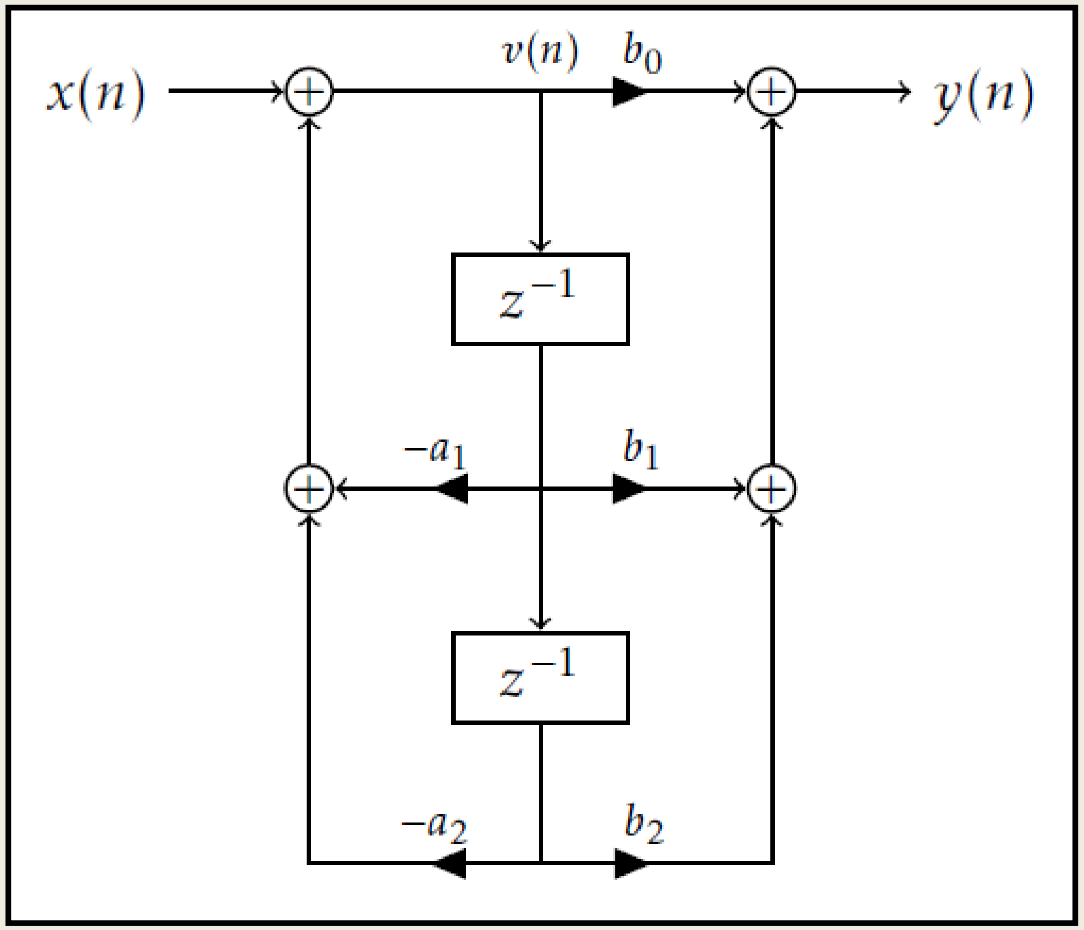
Number of delay elements = max (degree(numerator),degree(denominator))

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



Assume processor with clock frequency 1 MHz

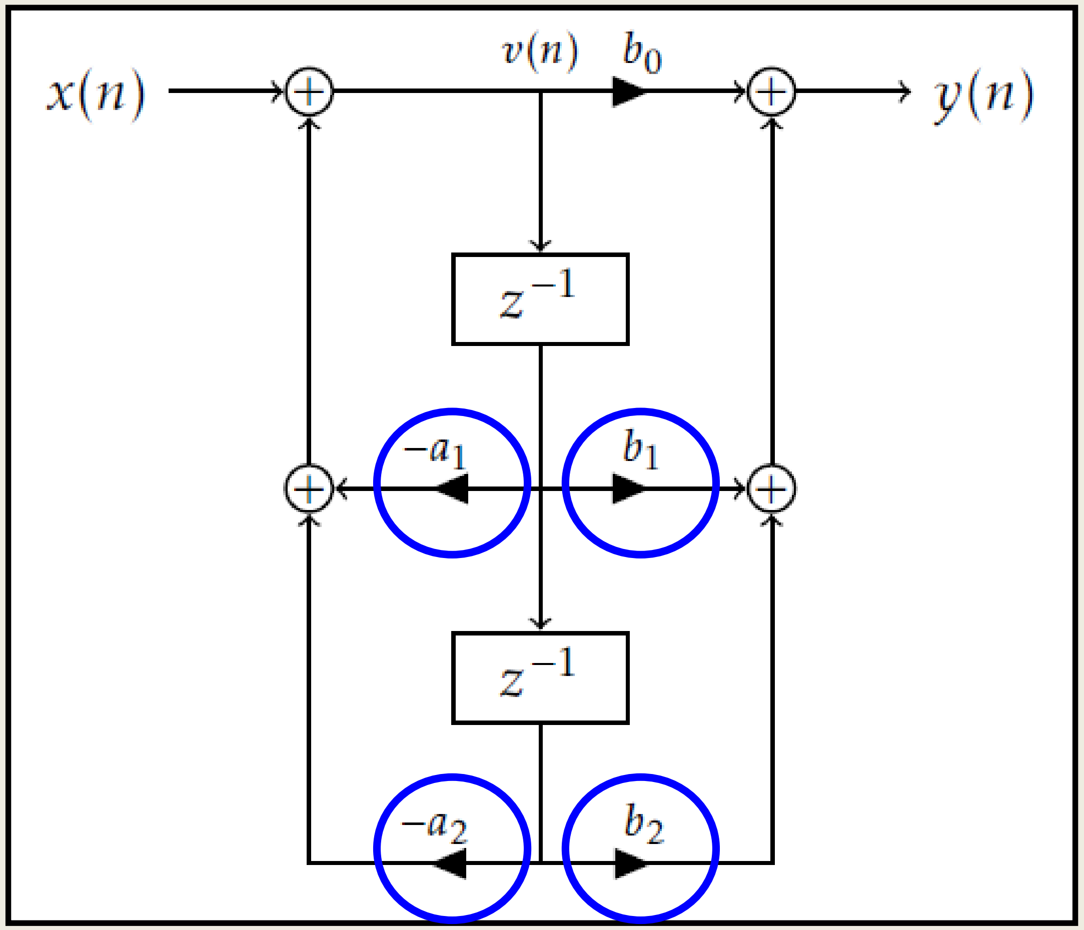
At what rate do we output y(n) ?

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ?

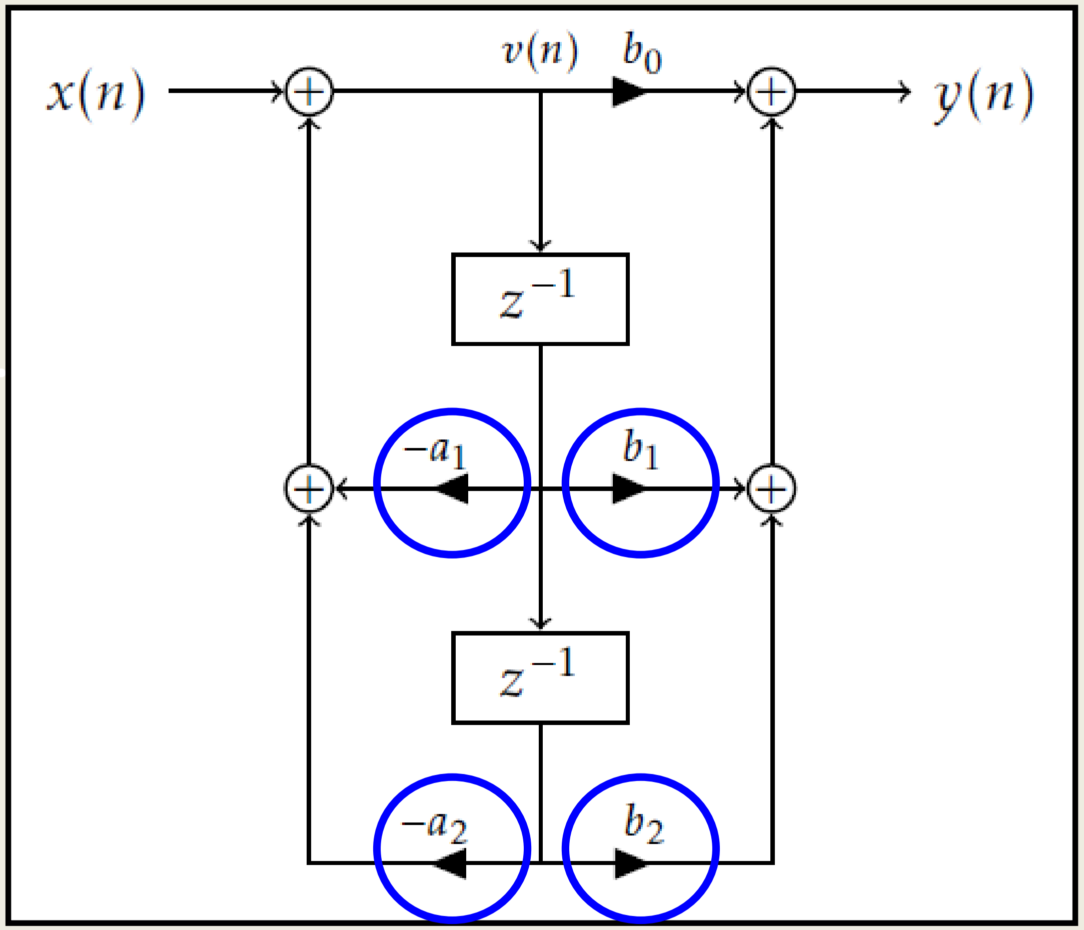
First we need to do parallell multiplications

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ?

First we need to do parallell multiplications

Clock cycles between updating memory elements:

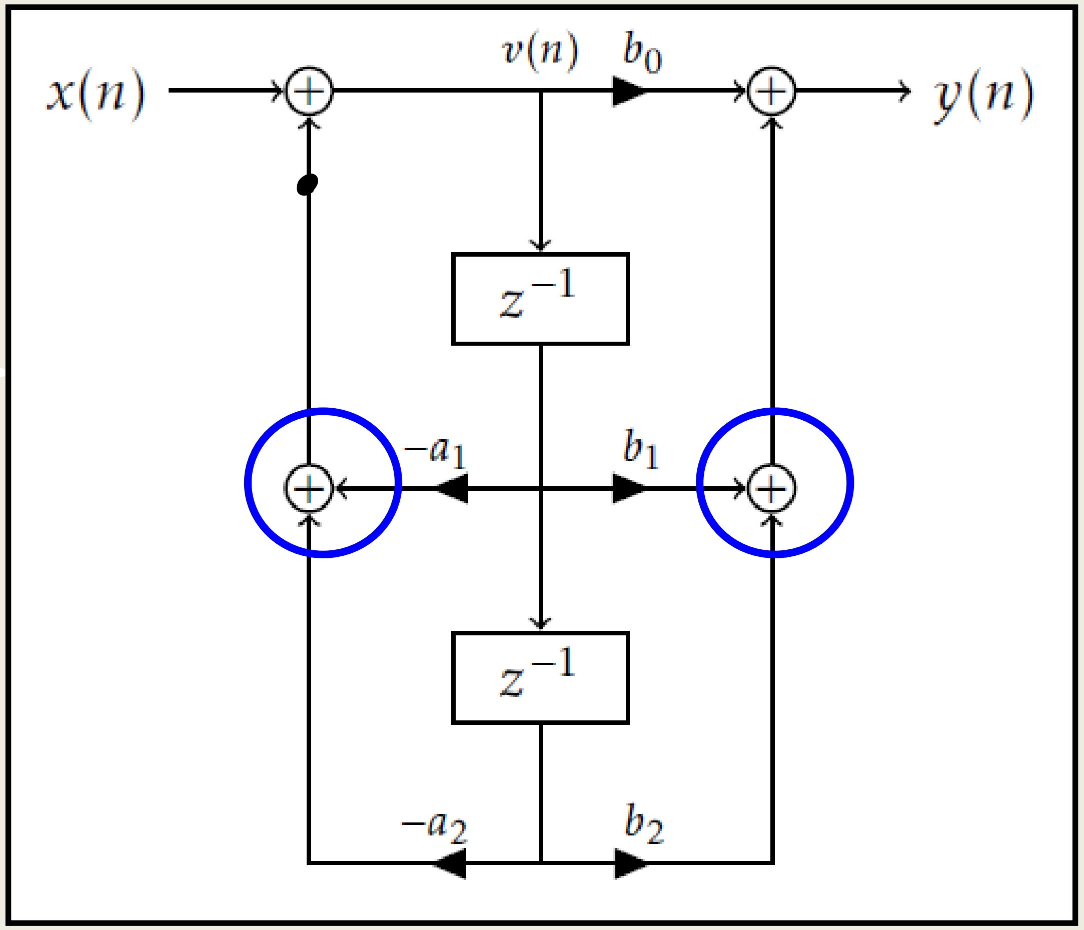
1 + …

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ?

First we need to do parallell multiplications

Then, can perform parallell additions

Clock cycles between updating memory elements:

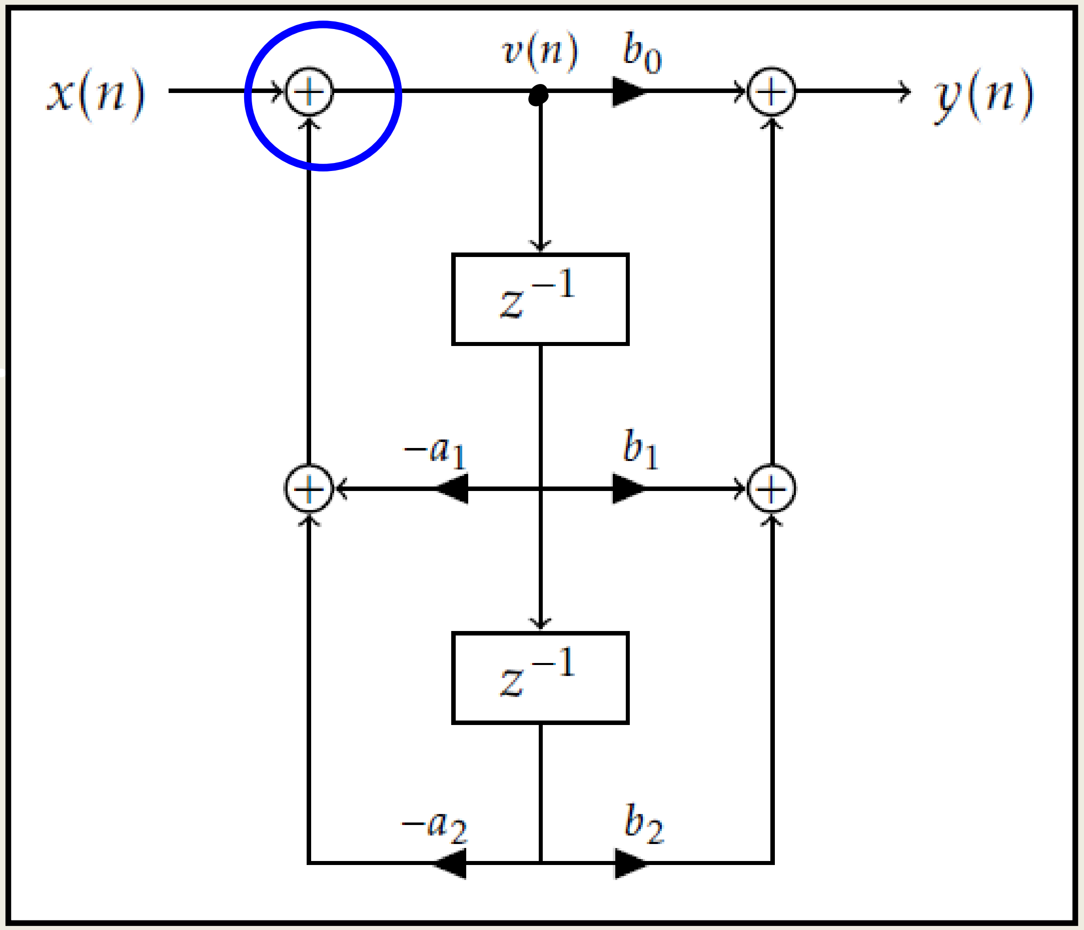
1 + 1 + …

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ?

First we need to do parallell multiplications

Then, can perform parallell additions

Then, another addition

Clock cycles between updating memory elements:

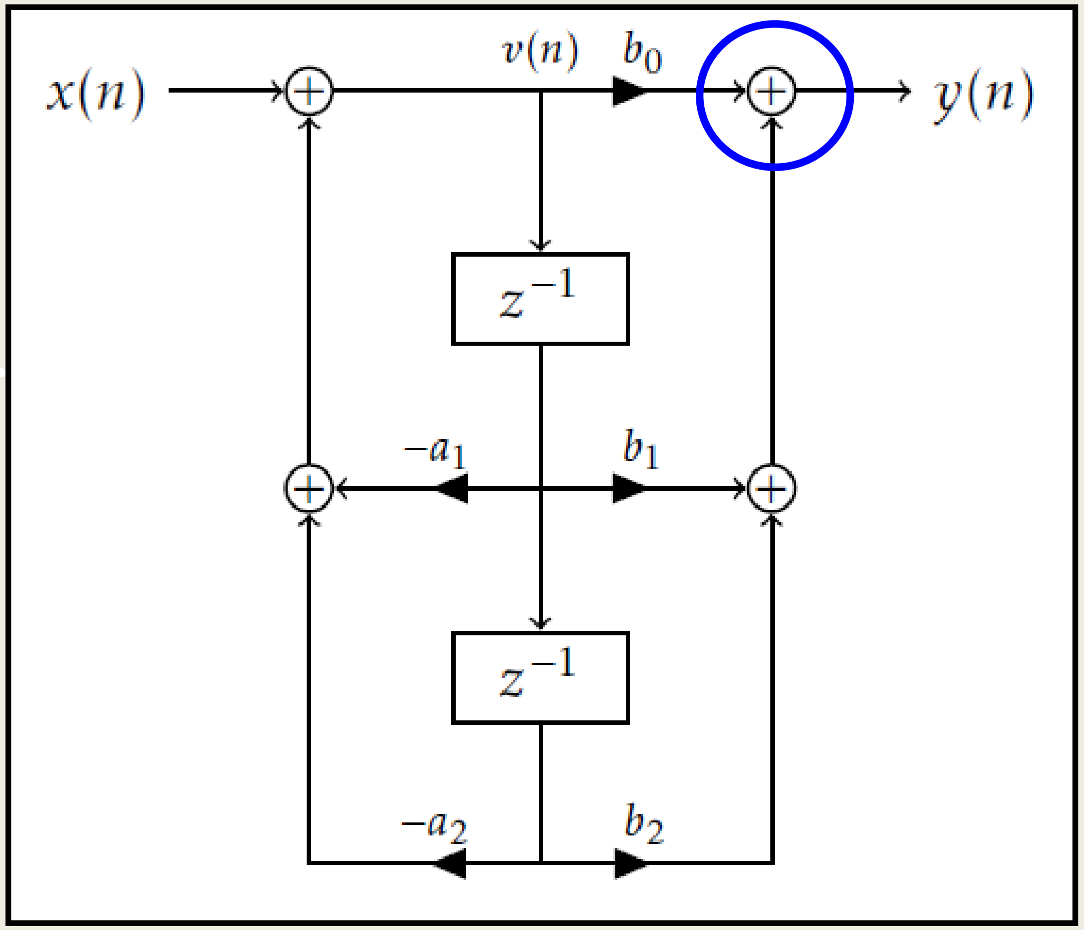
1 + 1 + 1 + …

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ?

First we need to do parallell multiplications

Then, can perform parallell additions

Then, another addition

Then one more

Clock cycles between updating memory elements:

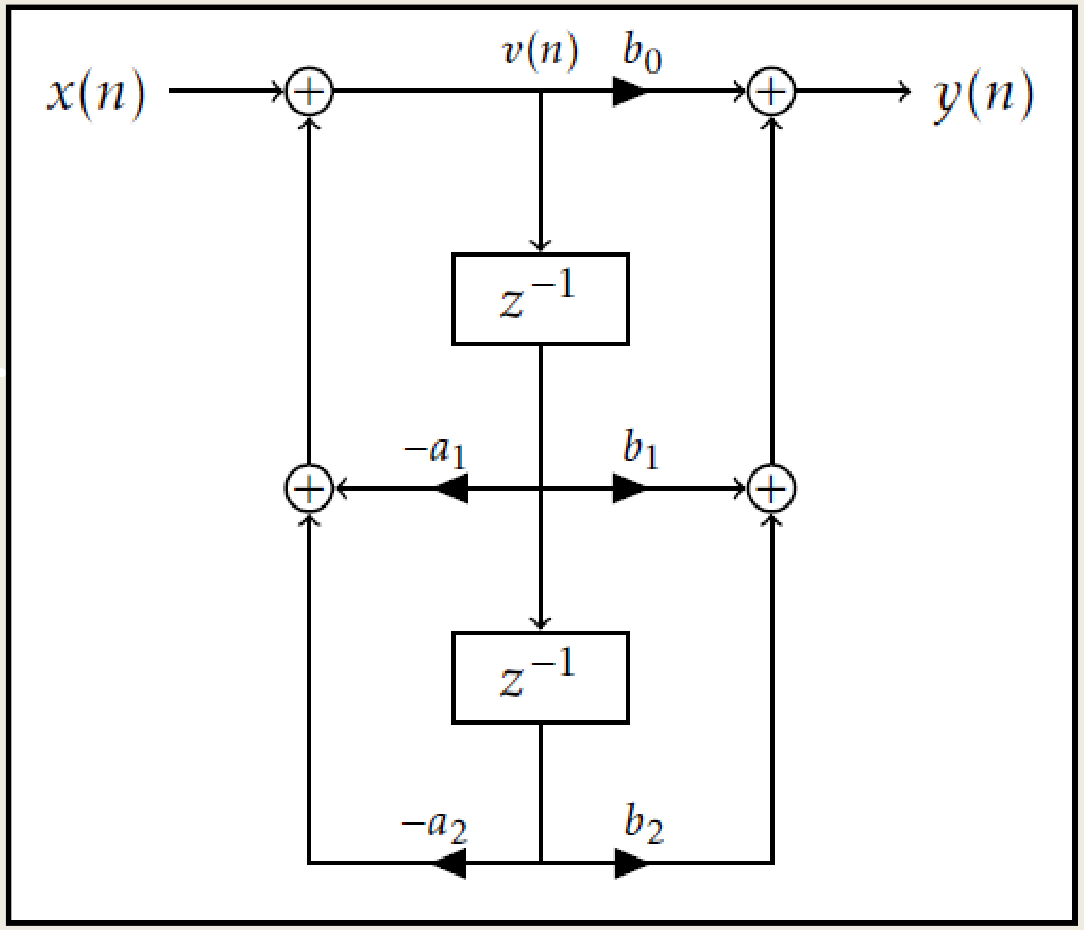
1 + 1 + 1 + 1 = 4

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ? 250kHz

Assuming 1 cycle per operation
(In reality, multiplications are more time consuming than additions.)

Clock cycles between updating memory elements:

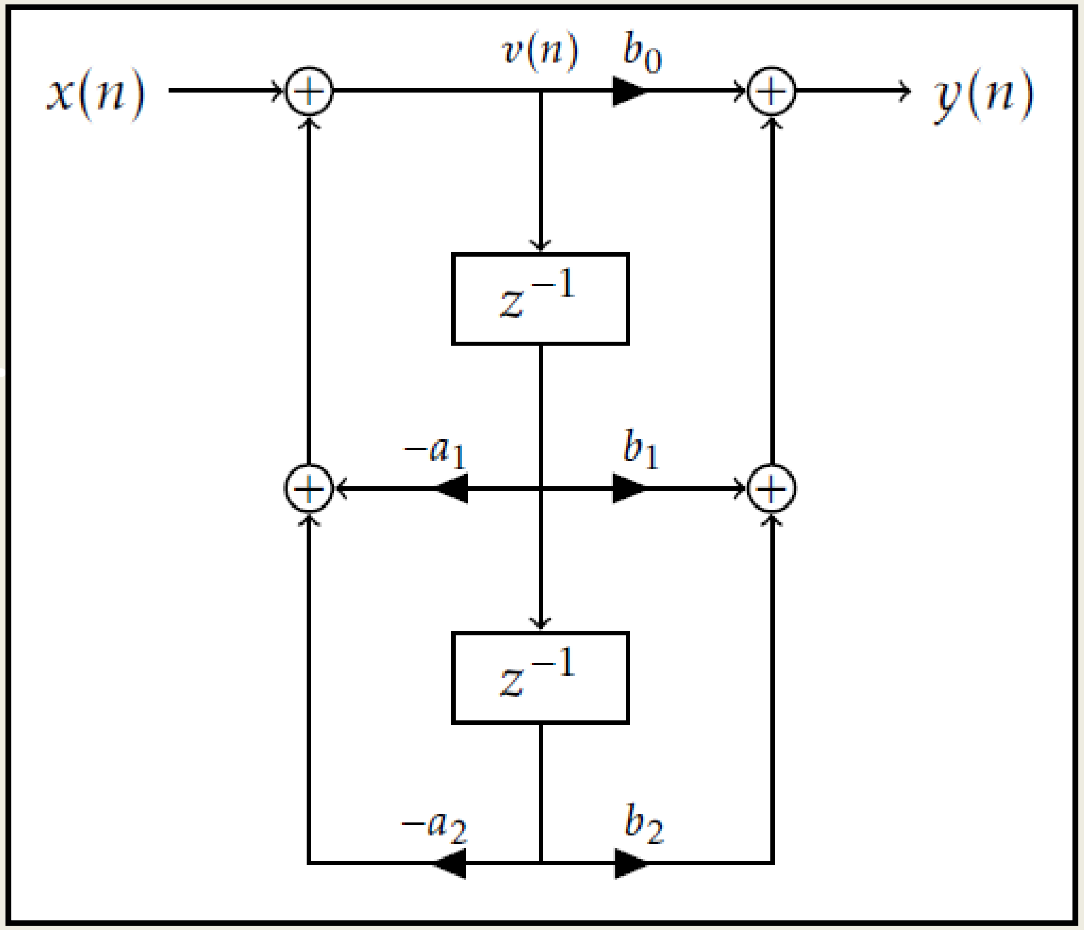
1 + 1 + 1 + 1 = 4

EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter



Transposition of systems:

For any block diagram, we obtain an equivalent if we,

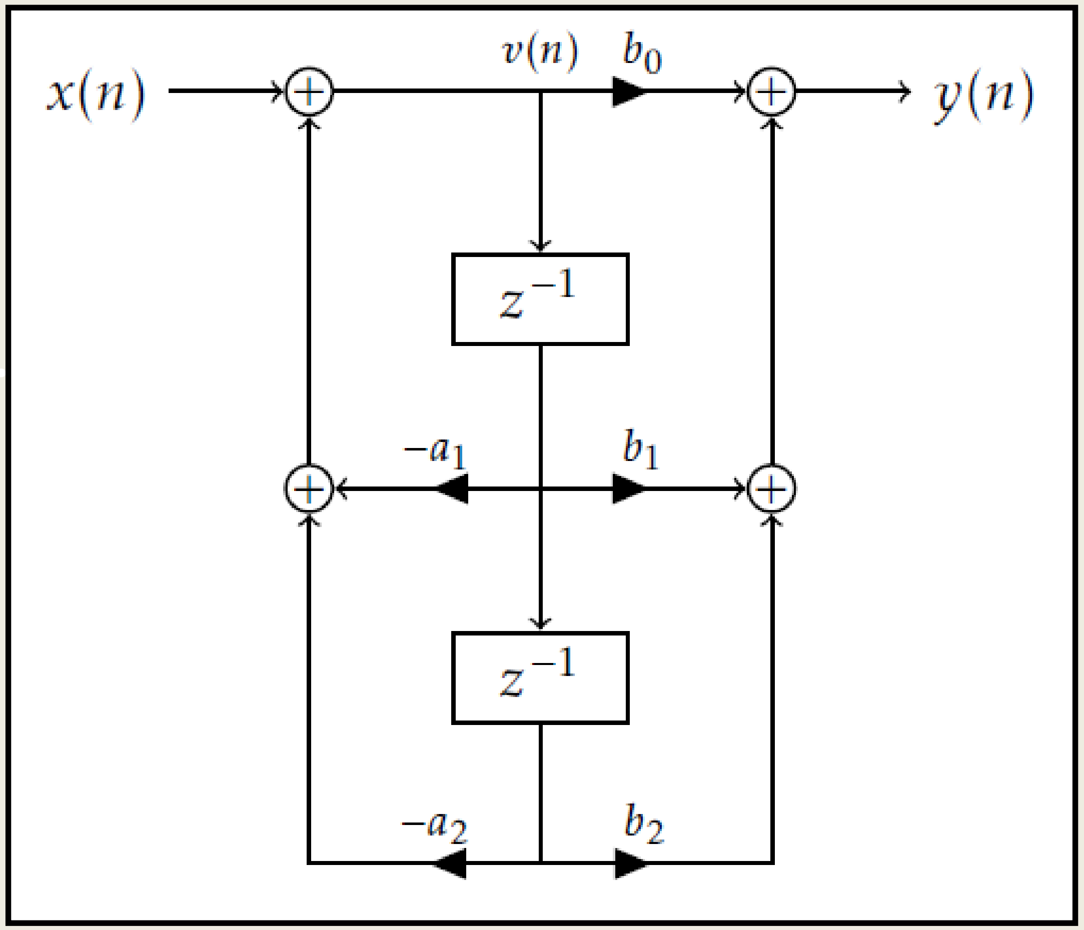
- Reverse direction of each interconnection
- Reverse direction of each multiplier
- Change junctions to adders and vice-versa
- Interchange input and output

EITF75 Systems and Signals

Some implementation aspects

IIR filters

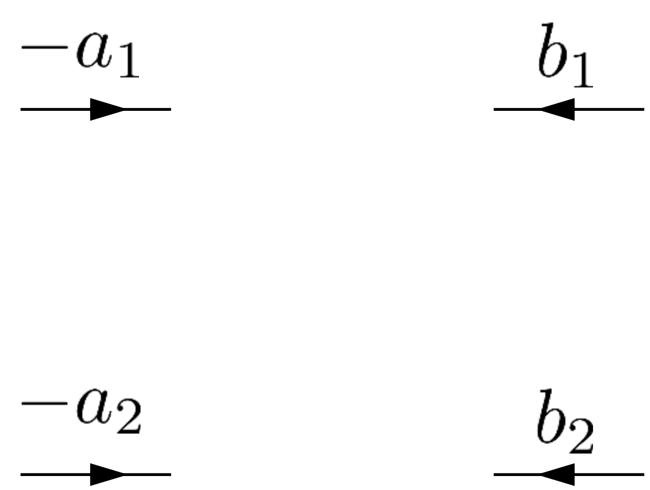
Second order filter



Transposition of systems:

For any block diagram, we obtain an equivalent if we,

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

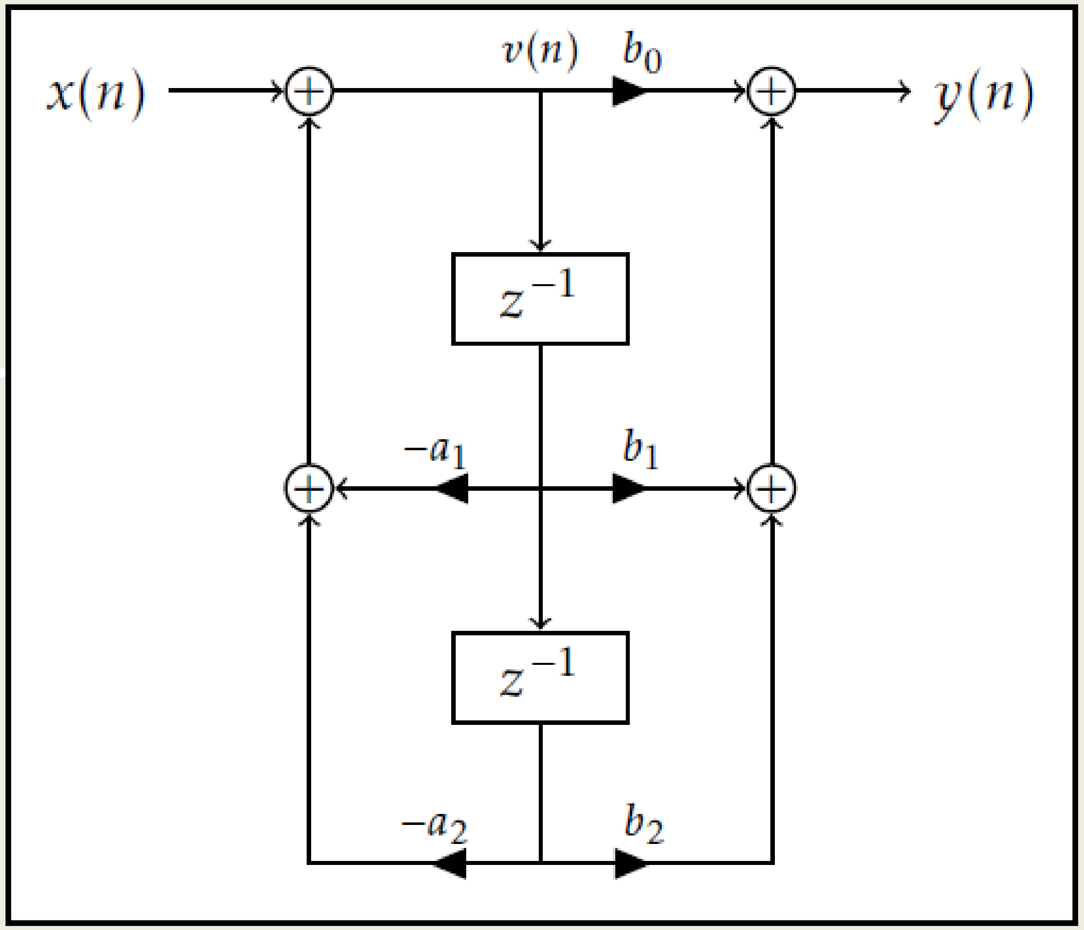


EITF75 Systems and Signals

Some implementation aspects

IIR filters

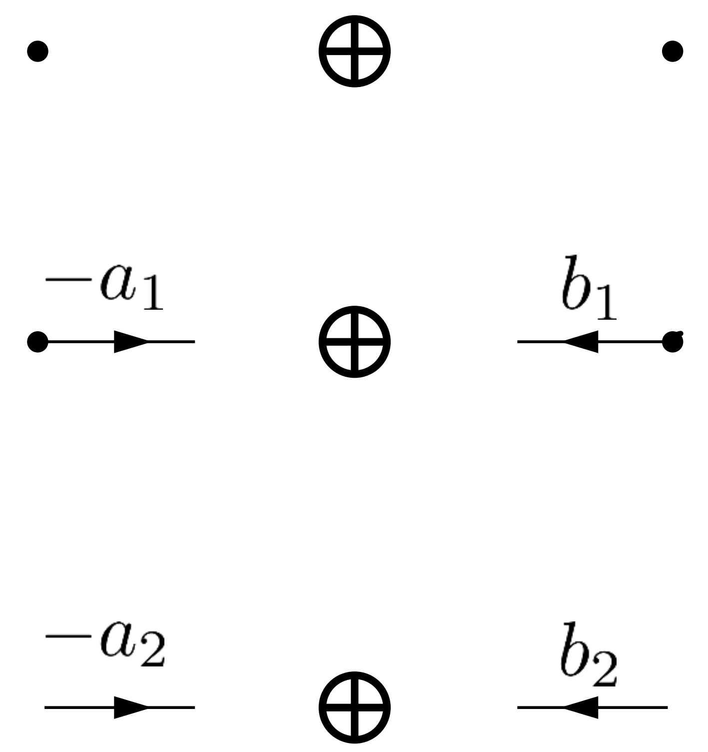
Second order filter



Transposition of systems:

For any block diagram, we obtain an equivalent if we,

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

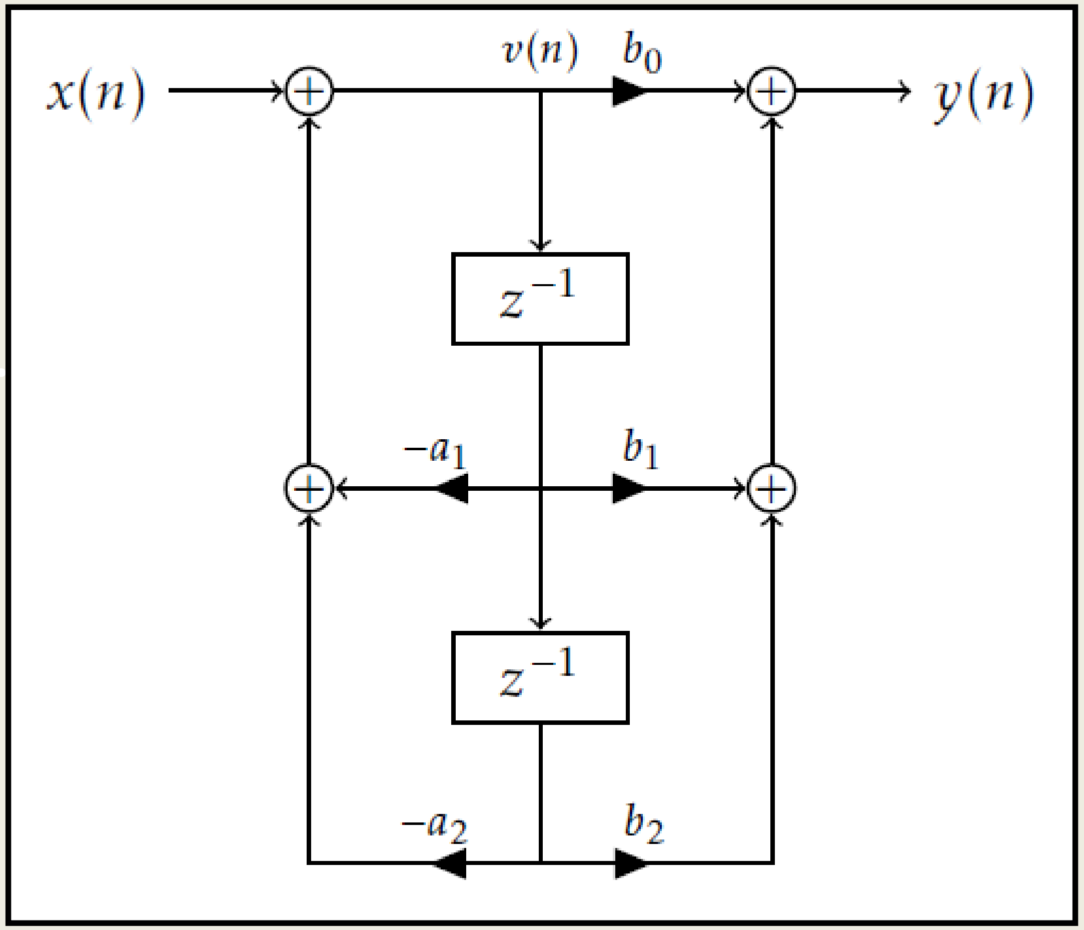


EITF75 Systems and Signals

Some implementation aspects

IIR filters

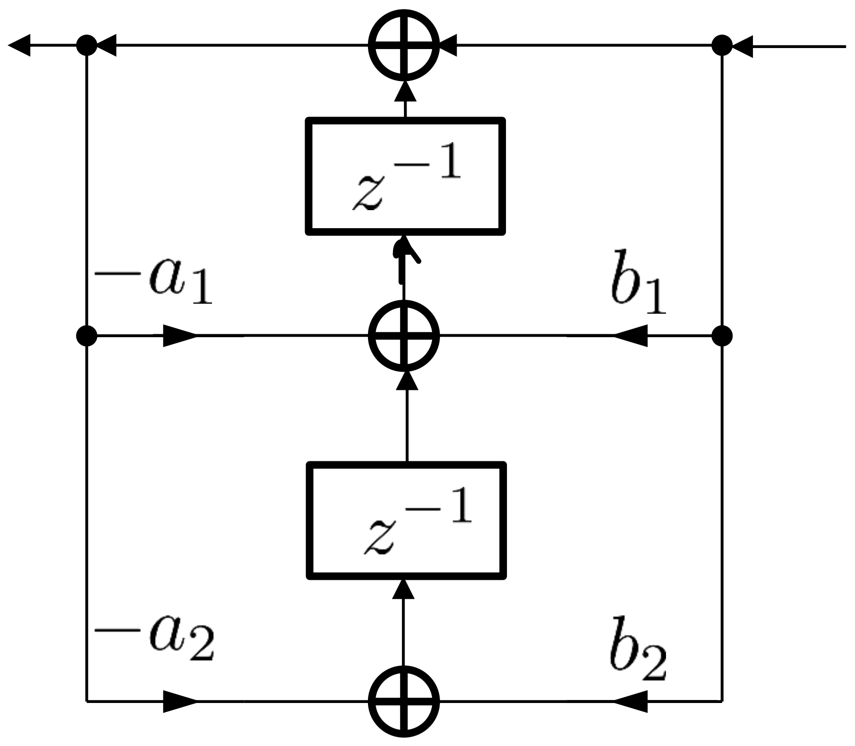
Second order filter



Transposition of systems:

For any block diagram, we obtain an equivalent if we,

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

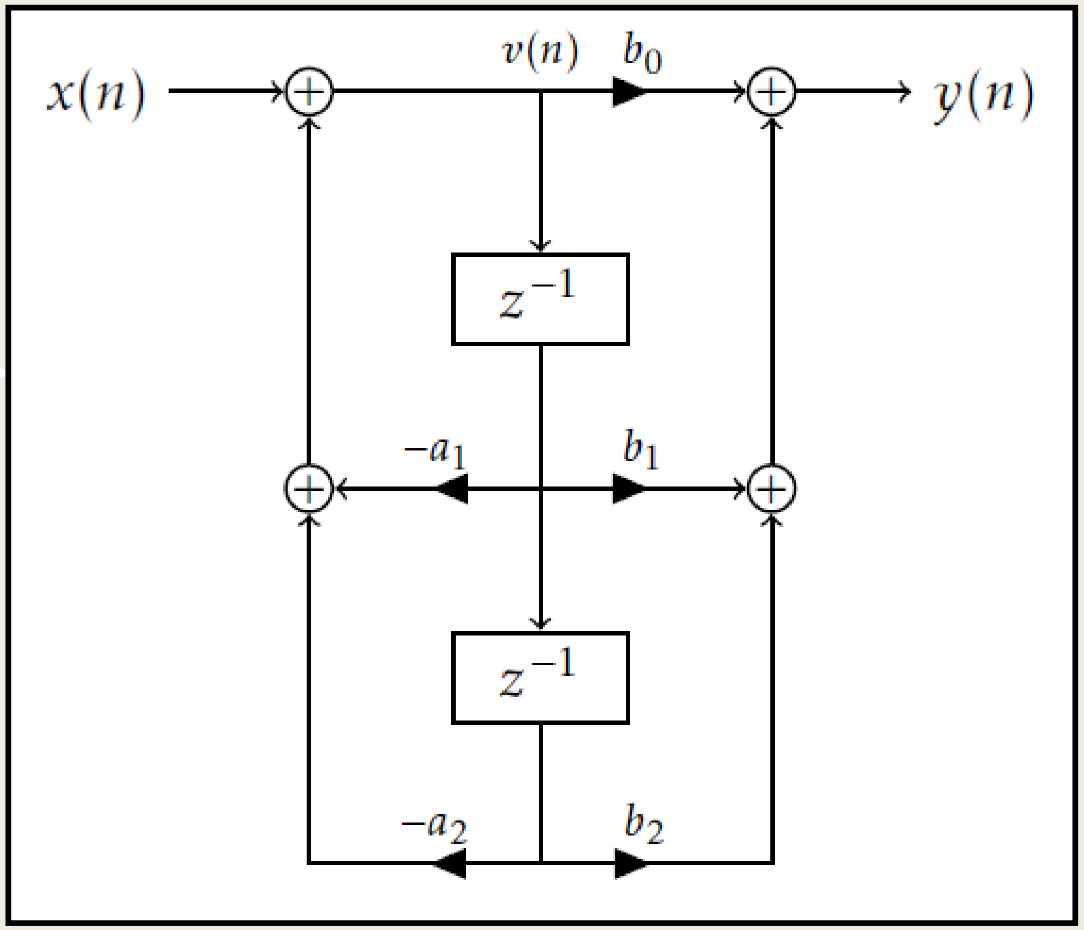


EITF75 Systems and Signals

Some implementation aspects

IIR filters

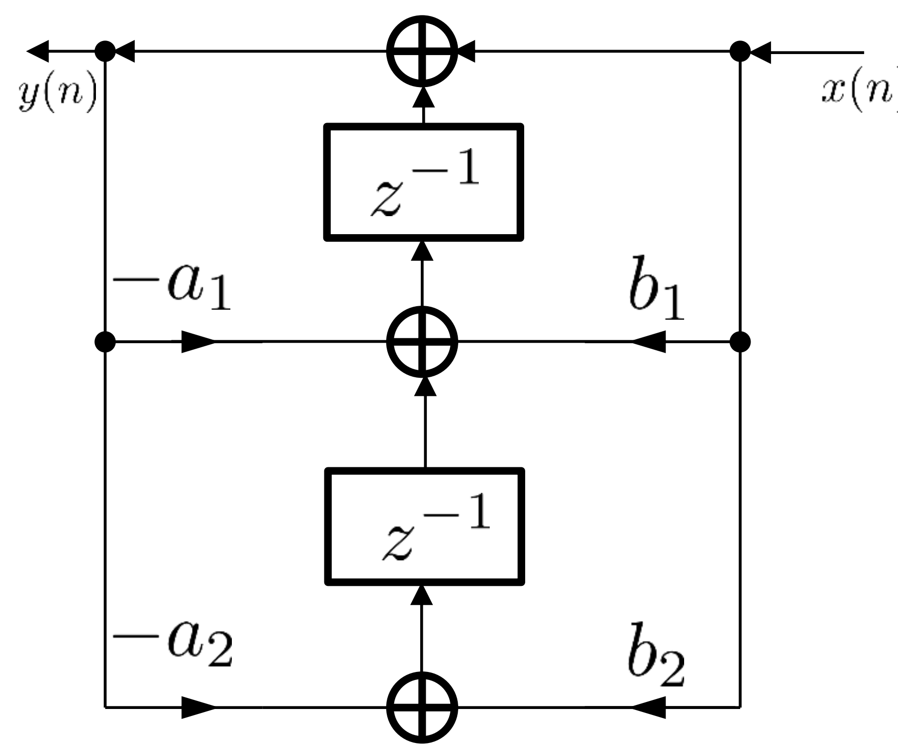
Second order filter



Transposition of systems:

For any block diagram, we obtain an equivalent if we,

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



EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter

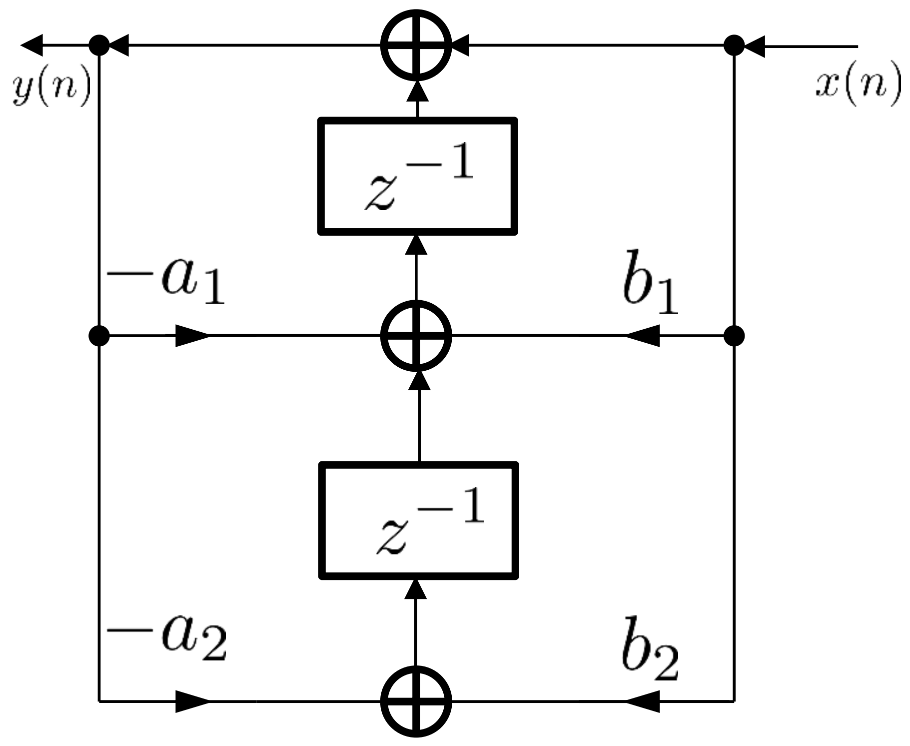
Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ?

Transposition of systems:

For any block diagram, we obtain an equivalent if we,

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



EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter

Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ?

First parallell multiplications

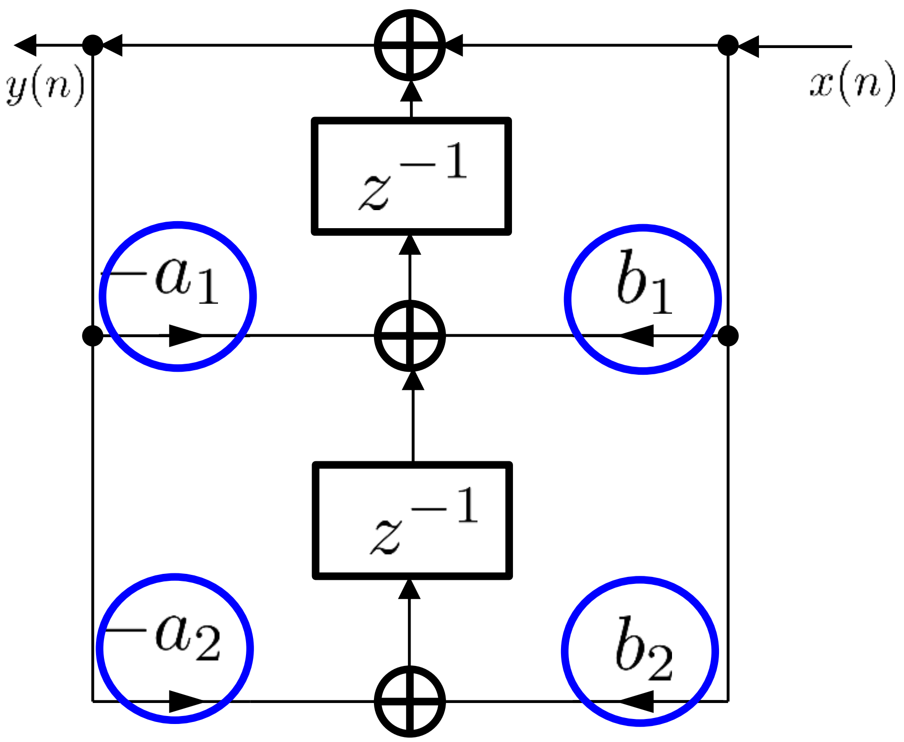
Clock cycles between updating memory elements:

1 + …

Transposition of systems:

For any block diagram, we obtain an equivalent if we,

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



EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter

Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ?

First parallell multiplications

Then parallell additions

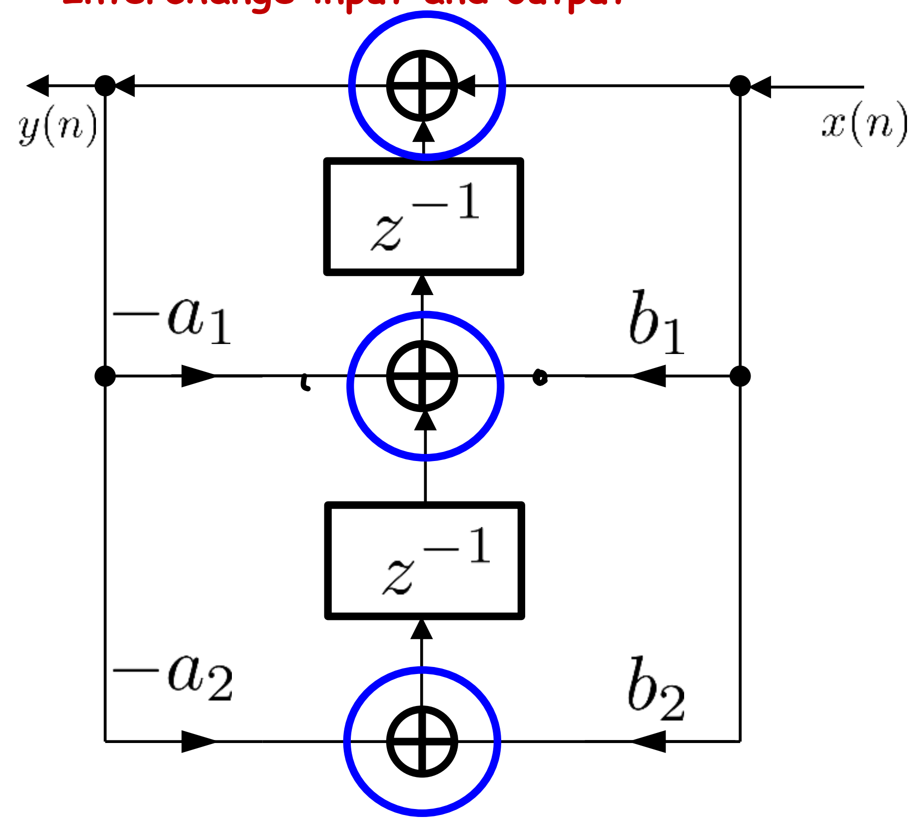
Clock cycles between updating memory elements:

1 + 1 + …

Transposition of systems:

For any block diagram, we obtain an equivalent if we,

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



EITF75 Systems and Signals

Some implementation aspects

IIR filters

Second order filter

Assume processor with clock frequency 1 MHz

At what rate do we output y(n) ? 500kHz

First parallell multiplications

Then parallell additions

Then done!

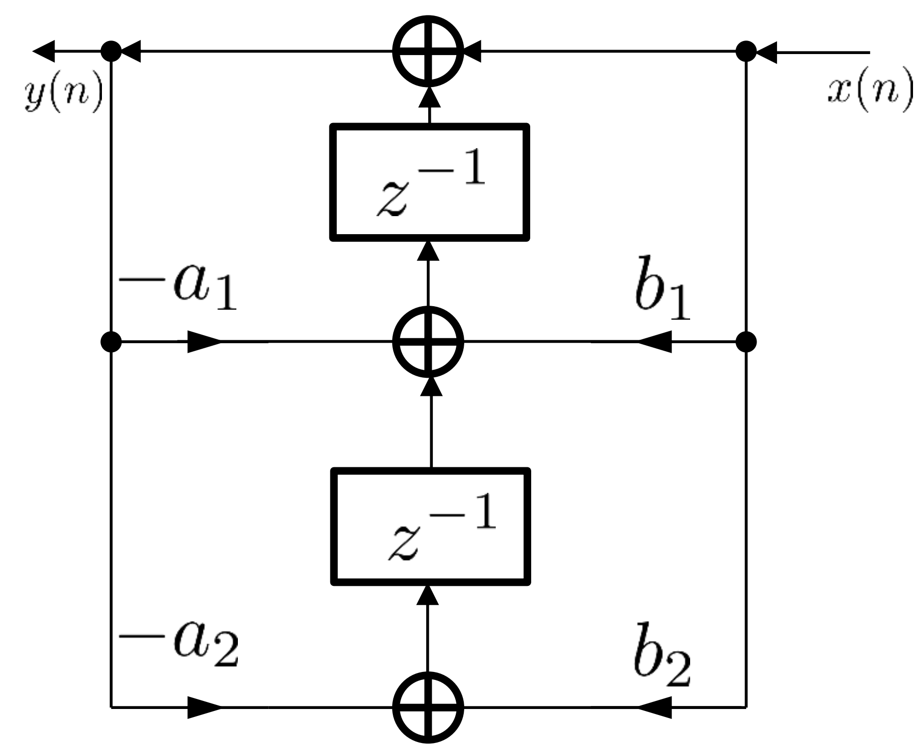
Clock cycles between updating memory elements:

1 + 1 = 2

Transposition of systems:

For any block diagram, we obtain an equivalent if we,

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



EITF75 Systems and Signals

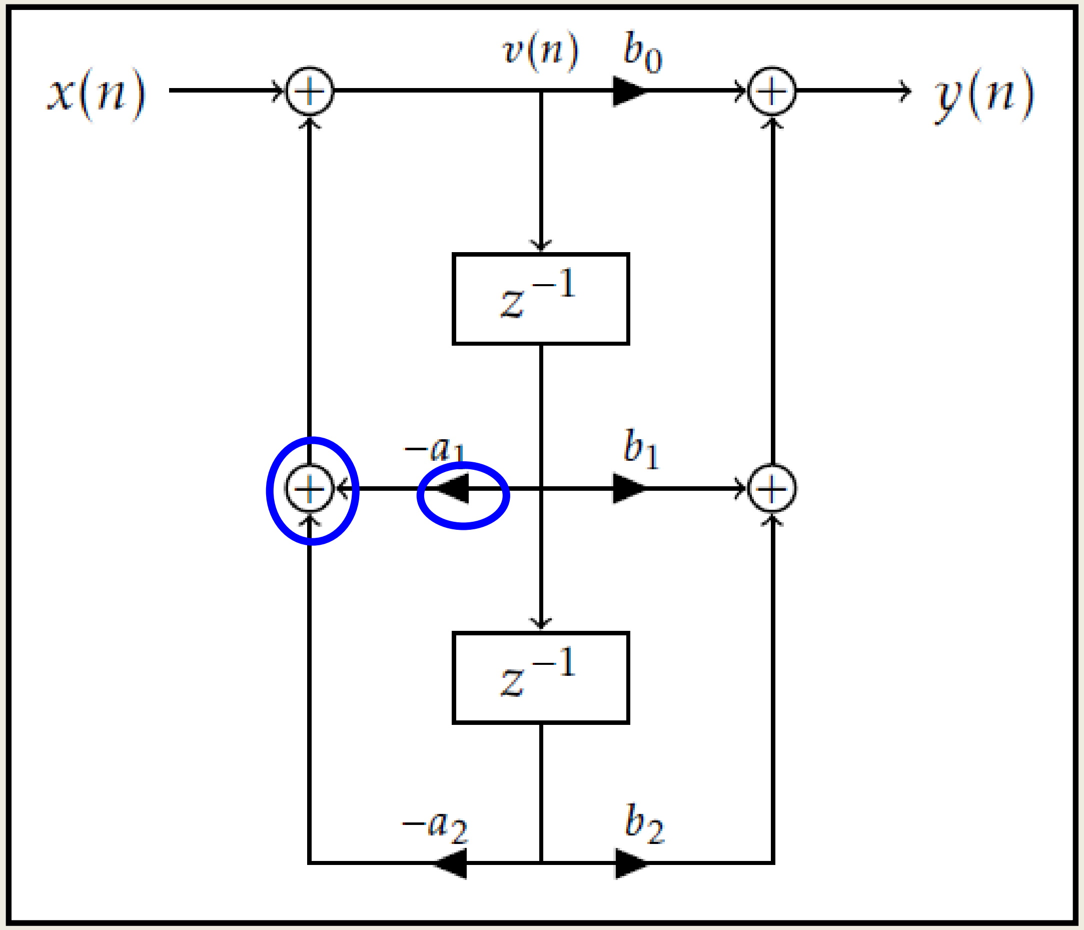
Some implementation aspects

Numerical precision issues

Coefficient precision: Coefficients are stored with finite precision. So implementation is not exact

Arithmetic precision: Done with finite precision, So not exact.

Typical model: Represent these effects as noise



EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues

Coefficient precision: Coefficients are stored with finite precision. So implementation is not exact

Arithmetic precision: Done with finite precision, So not exact.

Typical model: Represent these effects as noise

Example: Wilkinson's polynomial

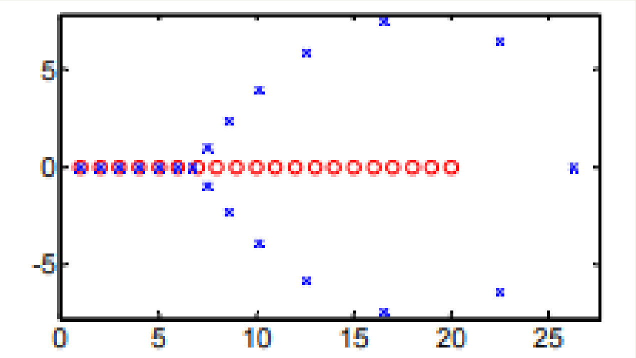
$$f(x) = \Pi_{n=1}^{20}(x-n) = x^{20} - 210x^{19} + 20615x^{18} - \dots$$

Zeros: on real axis, well separated

Assume imprecision: coefficient of x^{19} is 210.00021 (1.000001 times the real one)

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

Red: zeros of exact Wilkinson
Blue: zeros of imprecise Wilkinson



EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues

Consider a desired transfer function

$$H(z) = \frac{(z-z_1)(z-z_2)(z-z_3)\cdots(z-z_M)}{(z-p_1)(z-p_2)(z-p_3)\cdots(z-p_M)}$$

General rule: Not wise to implement this as a one-stage filter, i.e., using direct form II (or its transposed version)

Better solution. Typical case: each filter is second order ("biquad"). M' = M/2

$$\xrightarrow{x(n)} \boxed{H_1(z)} \longrightarrow \boxed{H_2(z)} \longrightarrow \boxed{H_3(z)} \longrightarrow \cdots \longrightarrow \boxed{H_{M'}(z)} \xrightarrow{y(n)}$$

Two questions:

1. Which poles to pair with which zeros ?
2. In which order should the filters appear ?

EITF75 Systems and Signals

Some implementation aspects

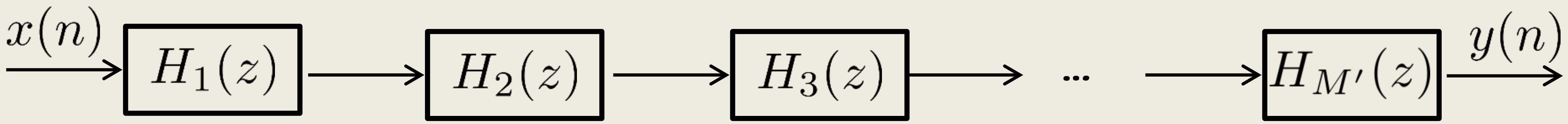
Numerical precision issues

Consider a desired transfer function

$$H(z) = \frac{(z-z_1)(z-z_2)(z-z_3)\cdots(z-z_M)}{(z-p_1)(z-p_2)(z-p_3)\cdots(z-p_M)}$$

General rule: Not wise to implement this as a one-stage filter, i.e., using direct form II (or its transposed version)

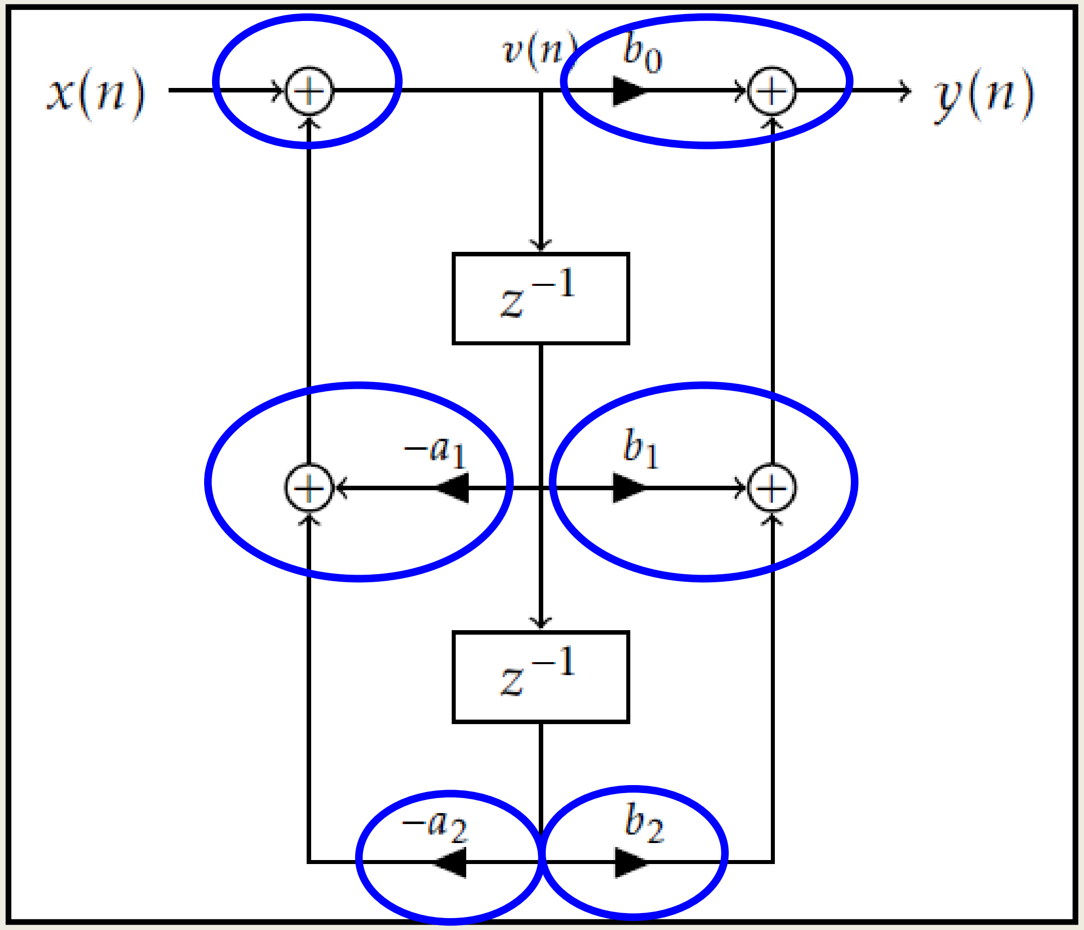
Better solution. Typical case: each filter is second order ("biquad"). M' = M/2



Model: Each filter produces noise that is being added to the input of itself

EITF75 Systems and Signals

Some implementation aspects

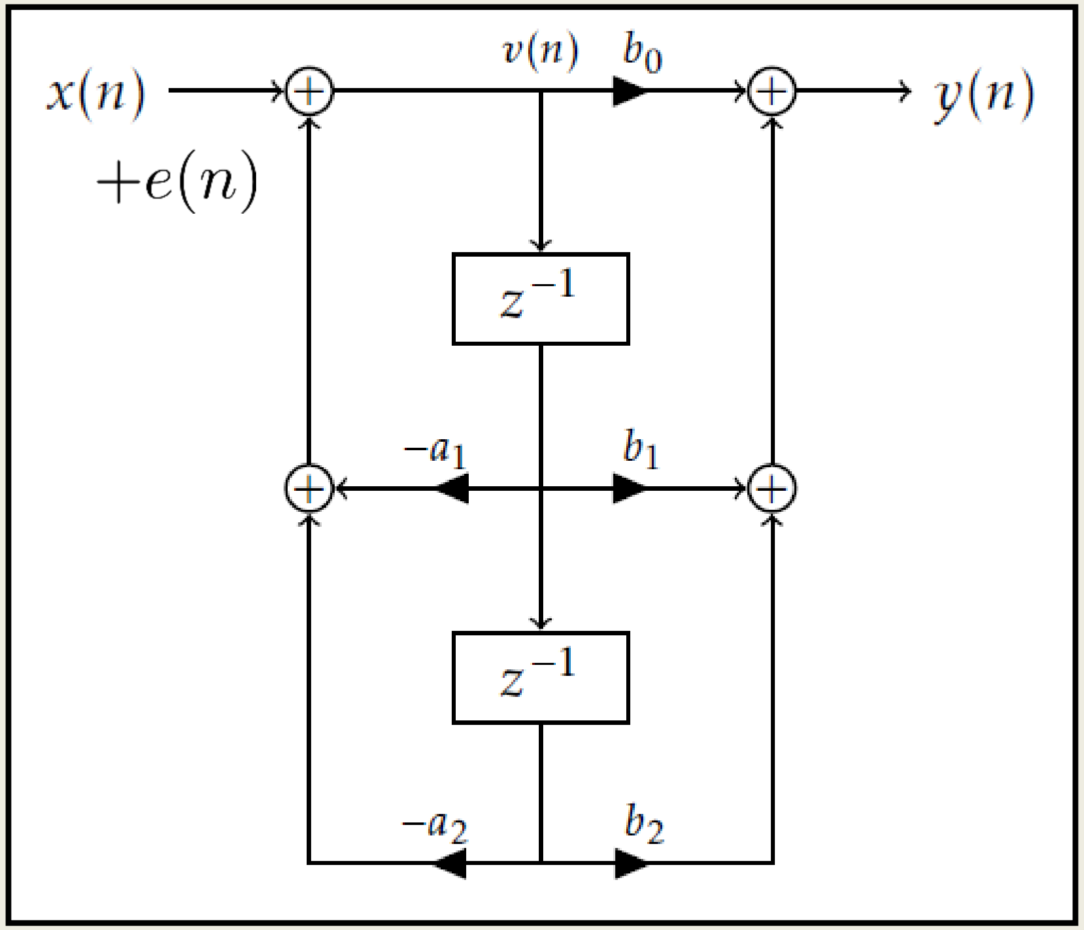


Rationale: The marked sections will not produce exact results

Model: Each filter produces noise that is being added to the input of itself

EITF75 Systems and Signals

Some implementation aspects



Rationale: The marked sections will not produce exact results
This can be modelled by adding a signal e(n) at the input

Model: Each filter produces noise that is being added to the input of itself

EITF75 Systems and Signals

Some implementation aspects

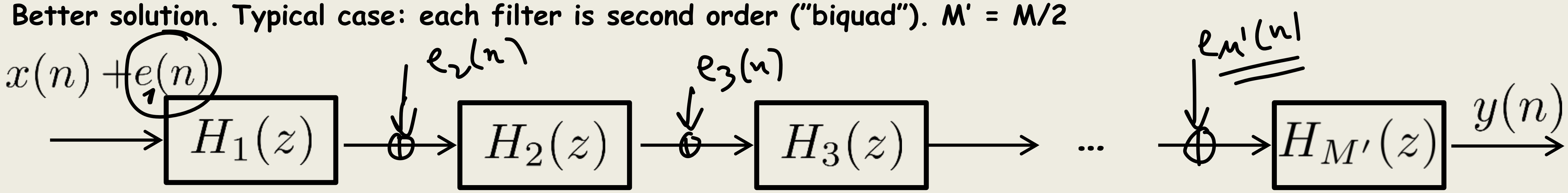
Numerical precision issues

Consider a desired transfer function

$$H(z) = \frac{(z-z_1)(z-z_2)(z-z_3)\cdots(z-z_M)}{(z-p_1)(z-p_2)(z-p_3)\cdots(z-p_M)}$$

General rule: Not wise to implement this as a one-stage filter, i.e., using direct form II (or its transposed version)

Better solution. Typical case: each filter is second order ("biquad"). M' = M/2



Model: Each filter produces noise that is being added to the input of itself

This noise will get amplified by later stages

$$H(z) = \frac{(z-z_1)(z-z_2)}{(z-p_1)(z-p_2)} \cdot \frac{(z-z_3)(z-z_4)}{(z-p_3)(z-p_4)} \cdot$$

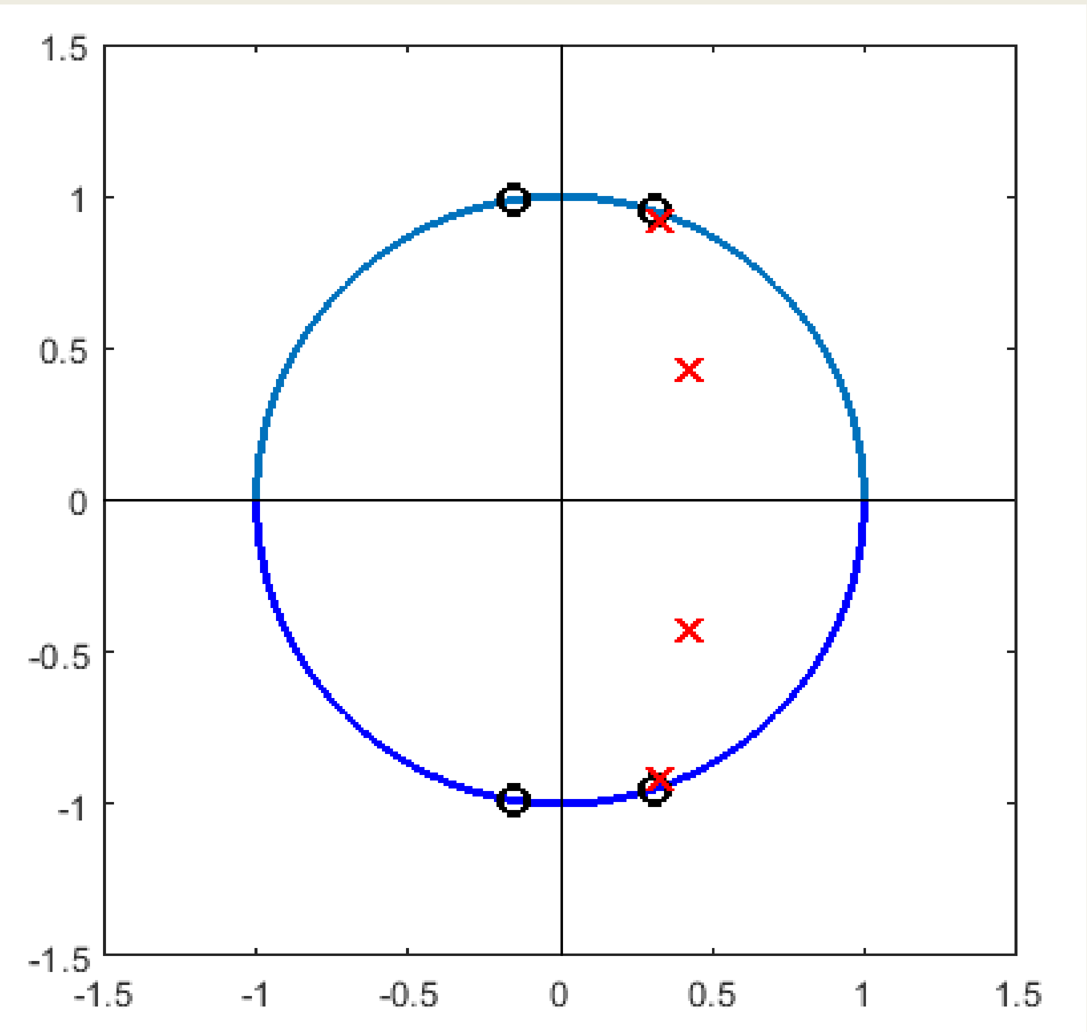
$$= H_1(z) \cdot H_2(z) \cdots$$

EITF75 Systems and Signals

Some implementation aspects

4 poles 4 zeros

Numerical precision issues



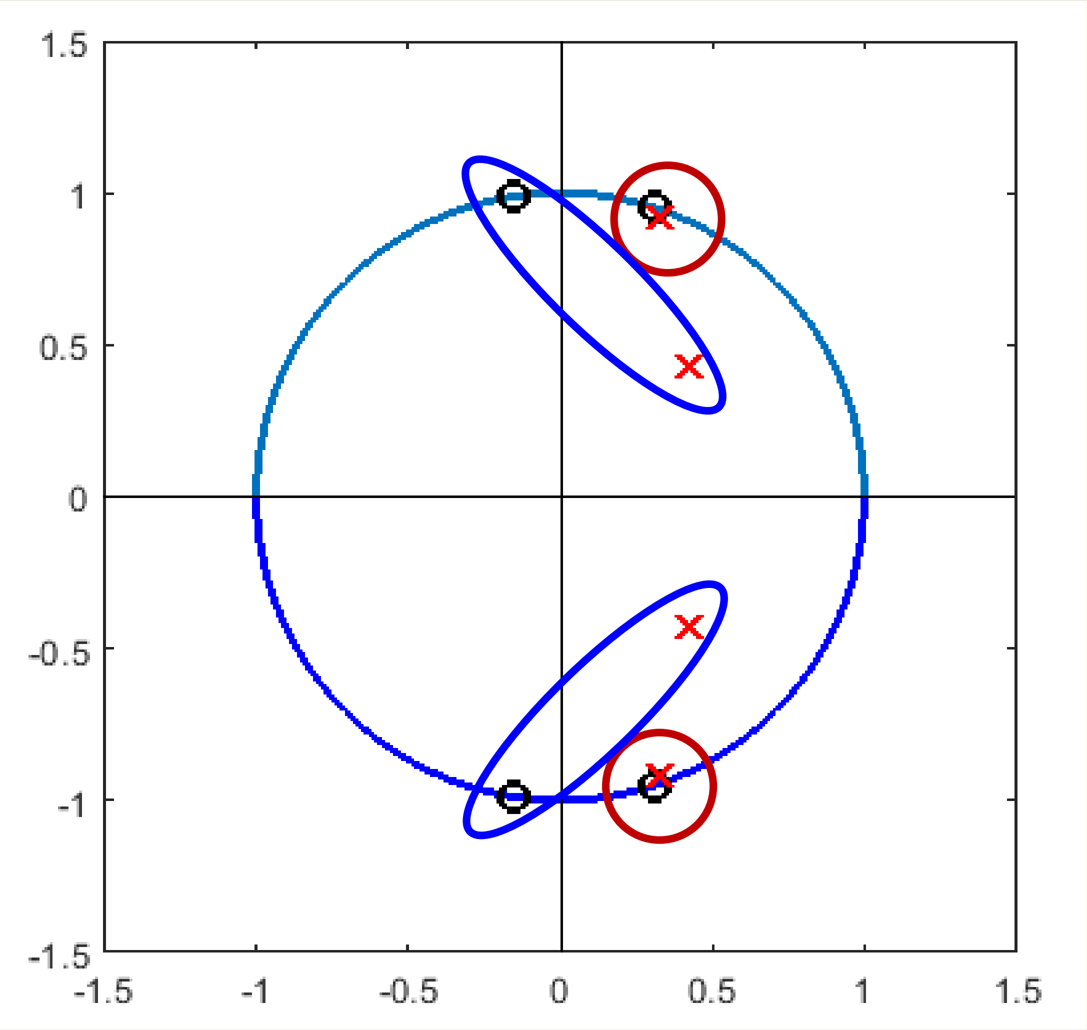
We have ~~to~~ two options for zero-pole combination

Consider the following filter

EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues



We have to options for zero-pole combination

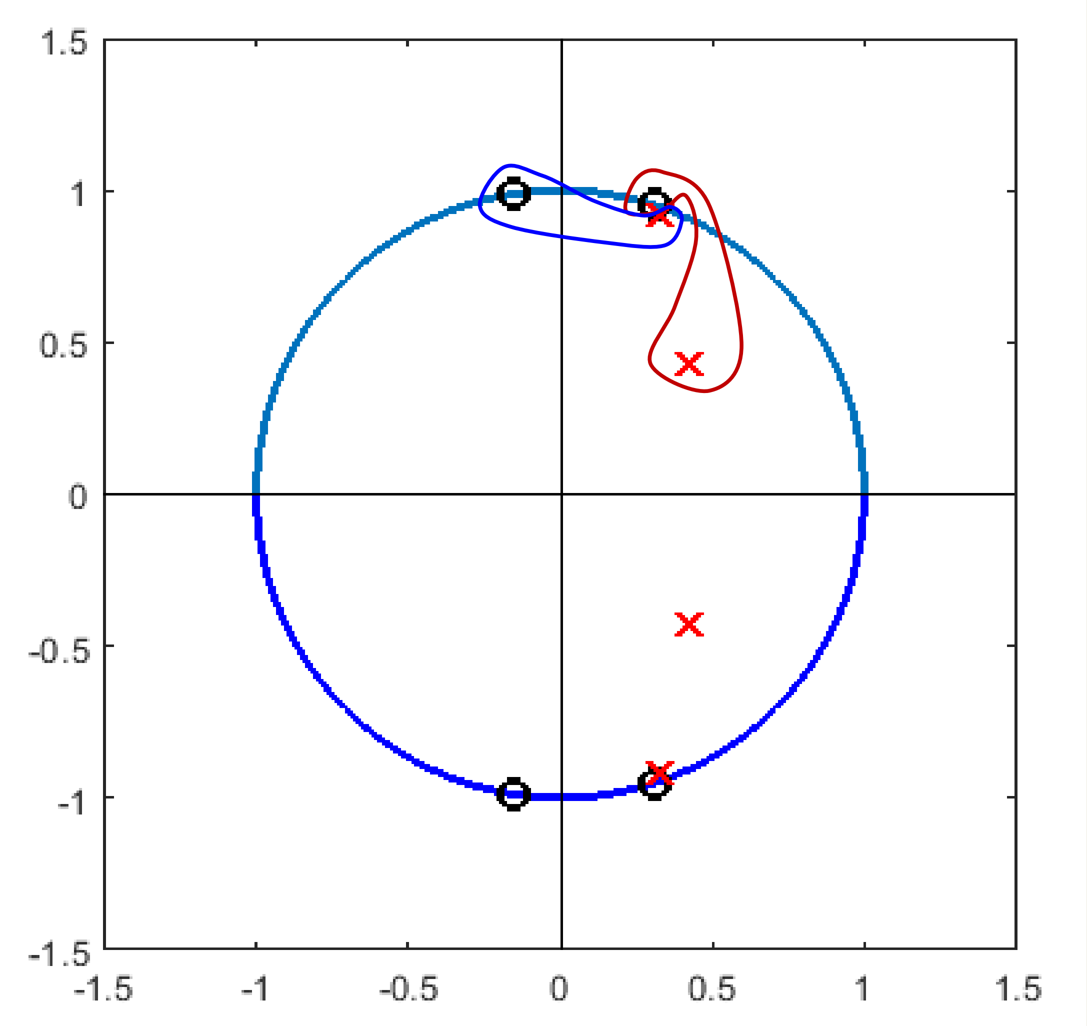
Option 1

Consider the following filter

EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues



We have to options for zero-pole combination

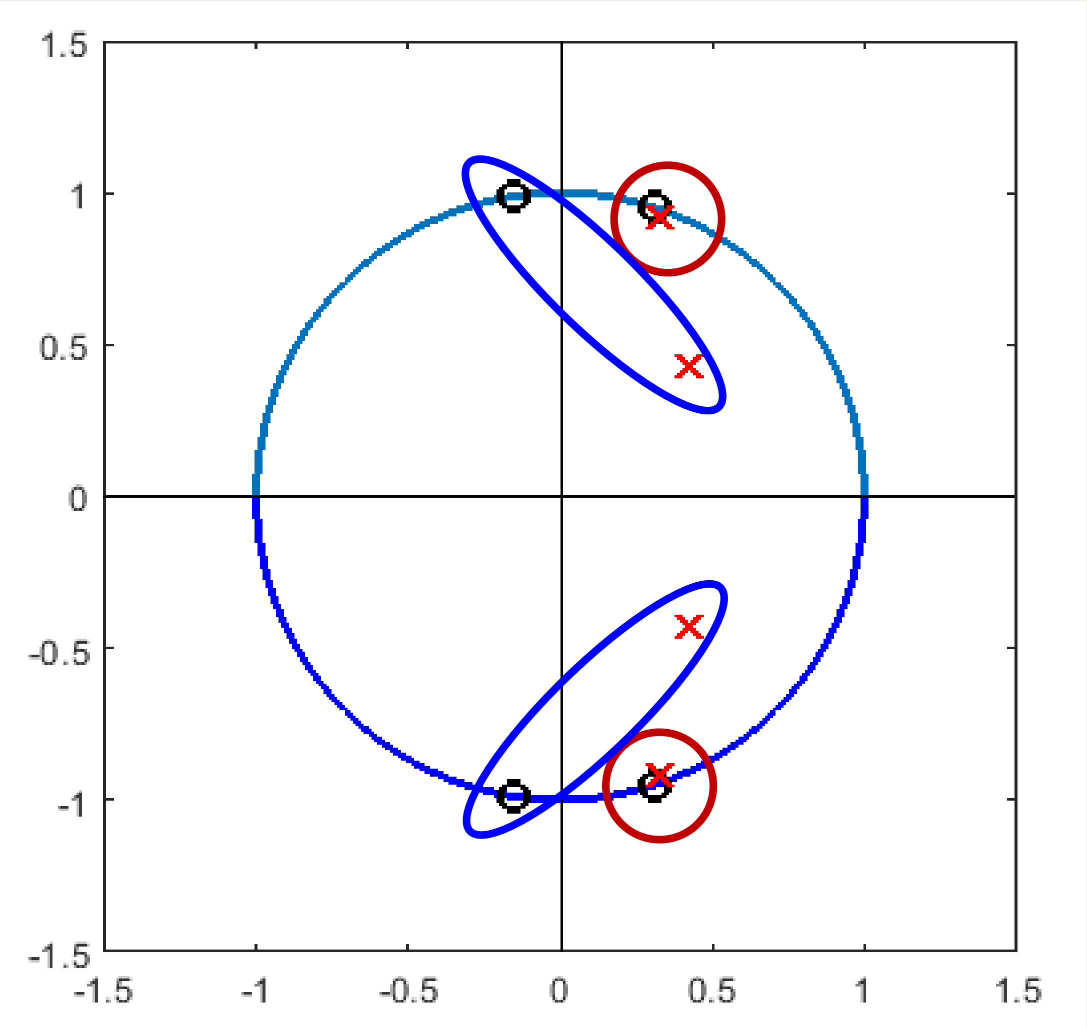
Option 2

Consider the following filter

EITF75 Systems and Signals

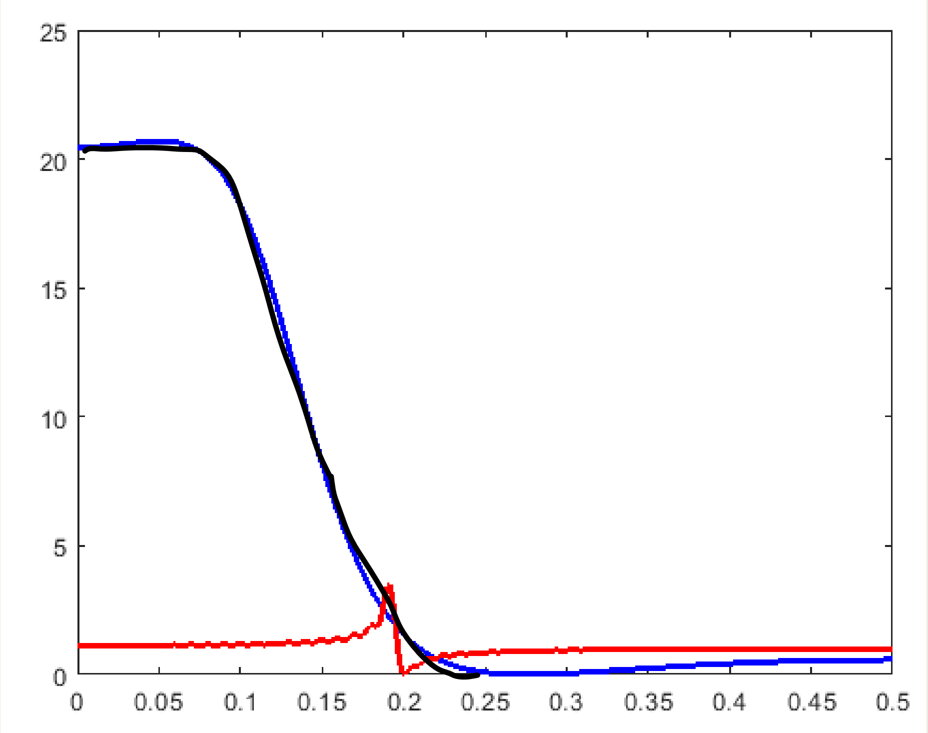
Some implementation aspects

Numerical precision issues



We have to options for zero-pole combination

Magnitude response



f

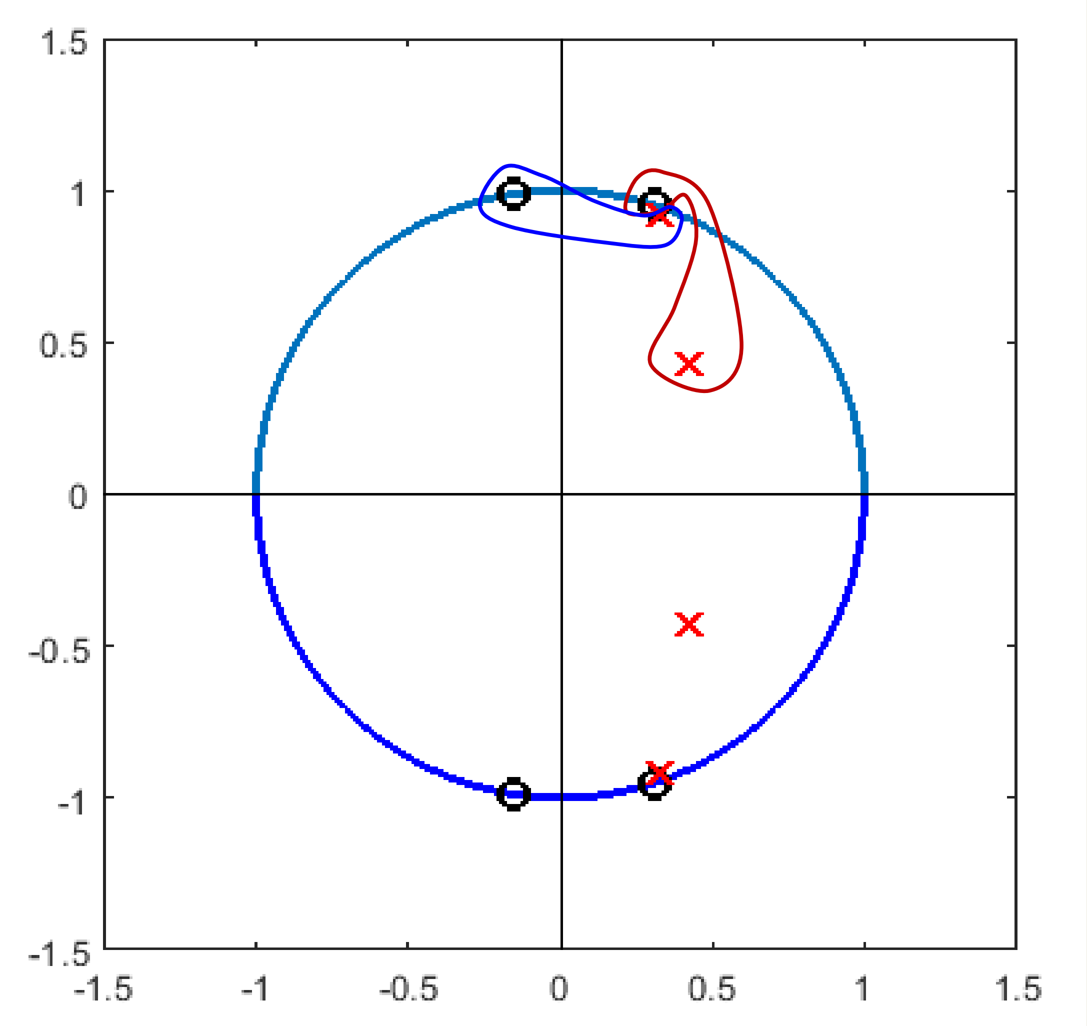
Option 1

Consider the following filter

EITF75 Systems and Signals

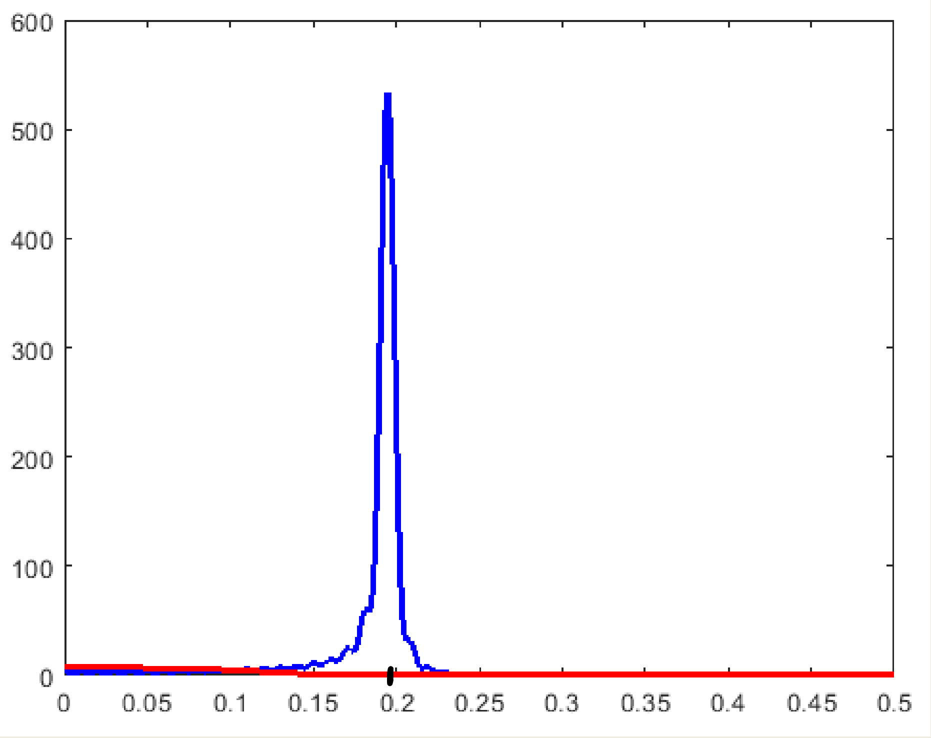
Some implementation aspects

Numerical precision issues



We have to options for zero-pole combination

Magnitude response



f

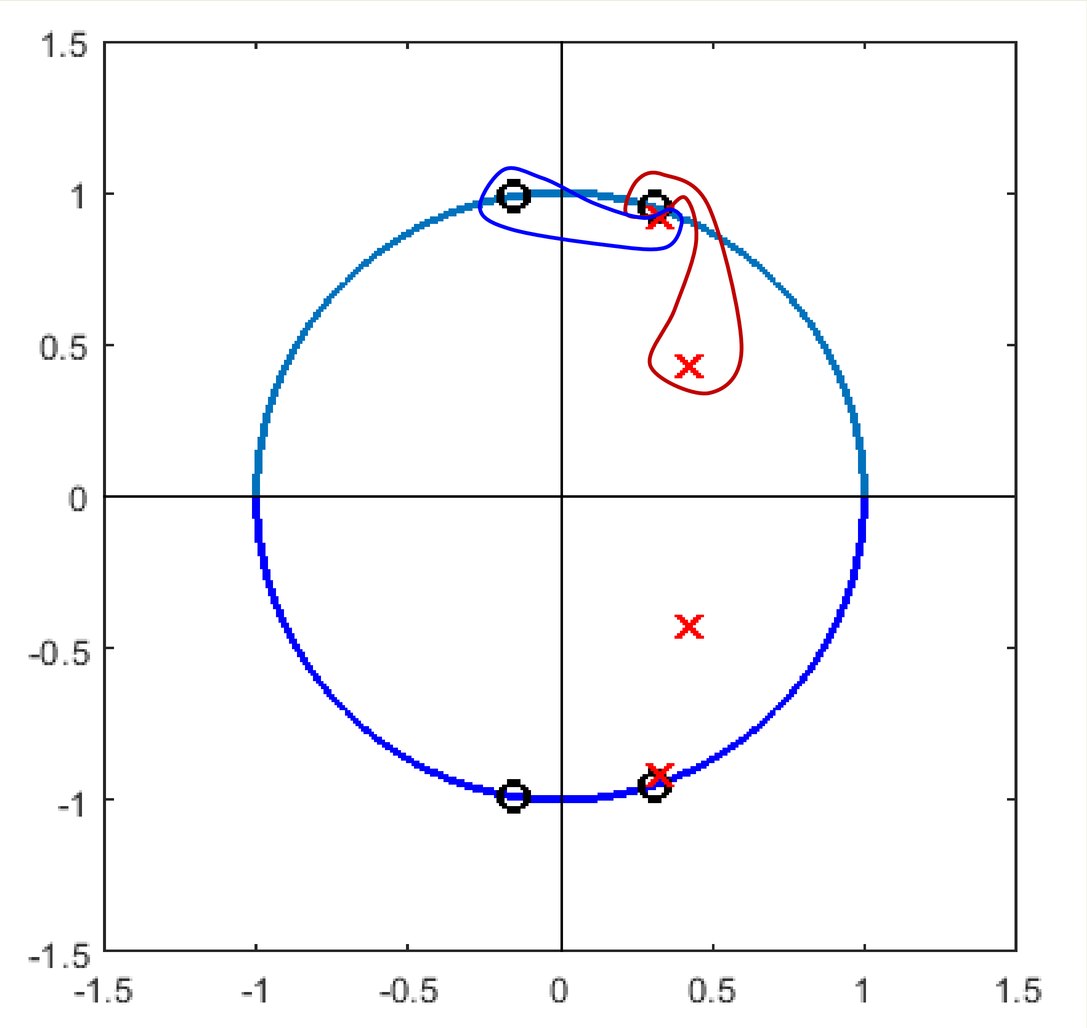
Option 2

Consider the following filter

EITF75 Systems and Signals

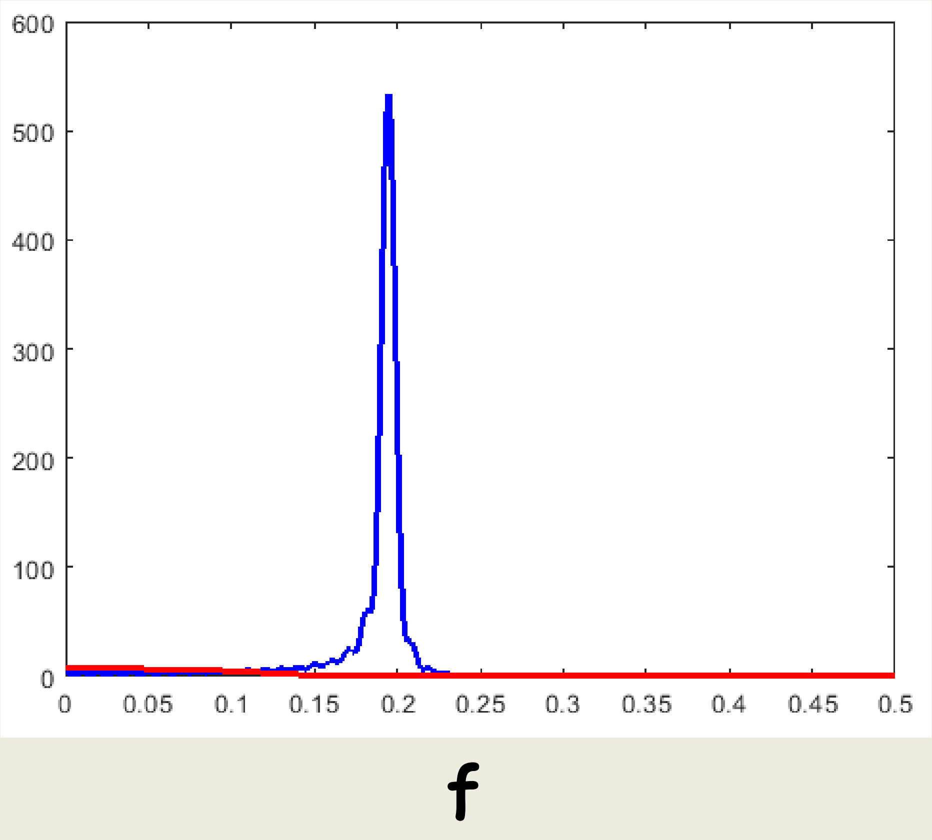
Some implementation aspects

Numerical precision issues



We have to options for zero-pole combination

Magnitude response



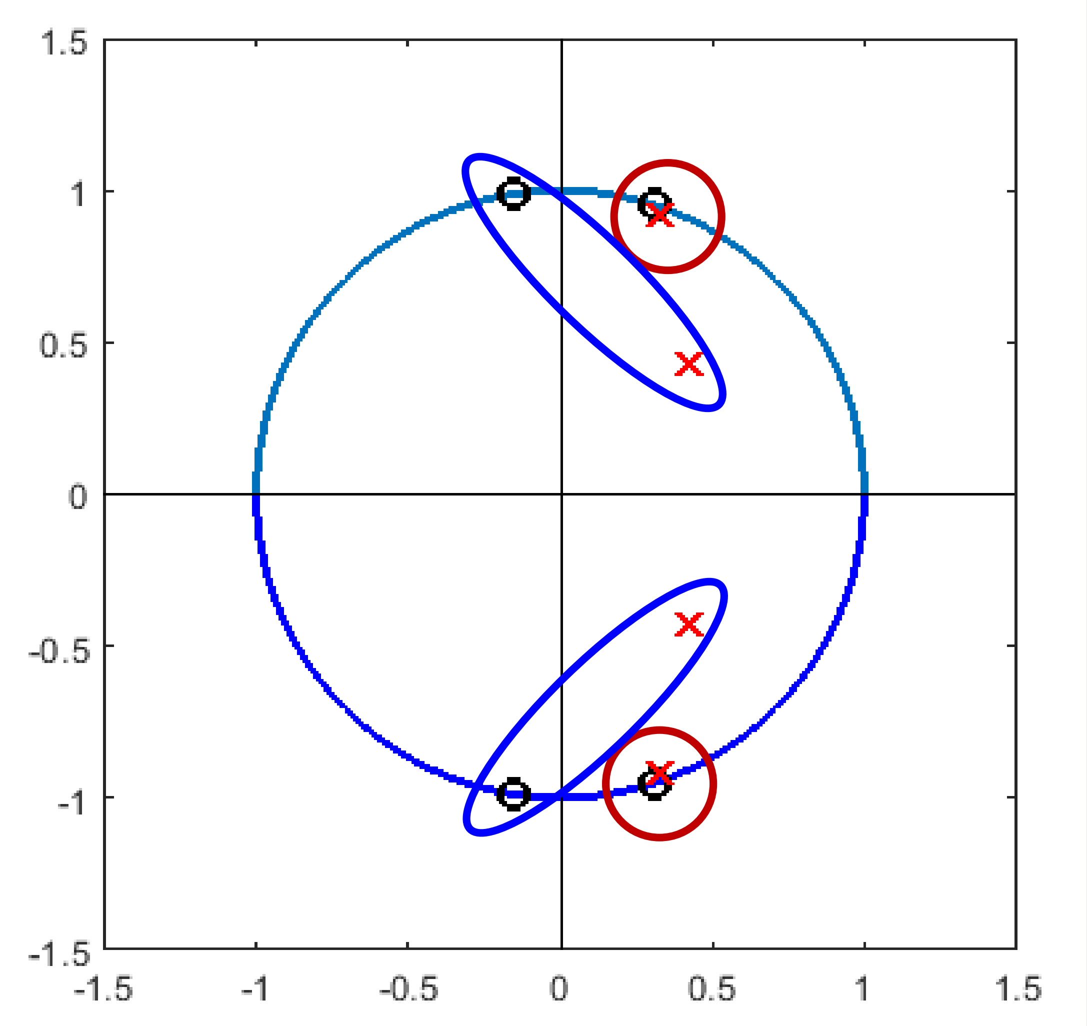
Option 2

This option would greatly amplify any source of noise we have. Not suitable.

EITF75 Systems and Signals

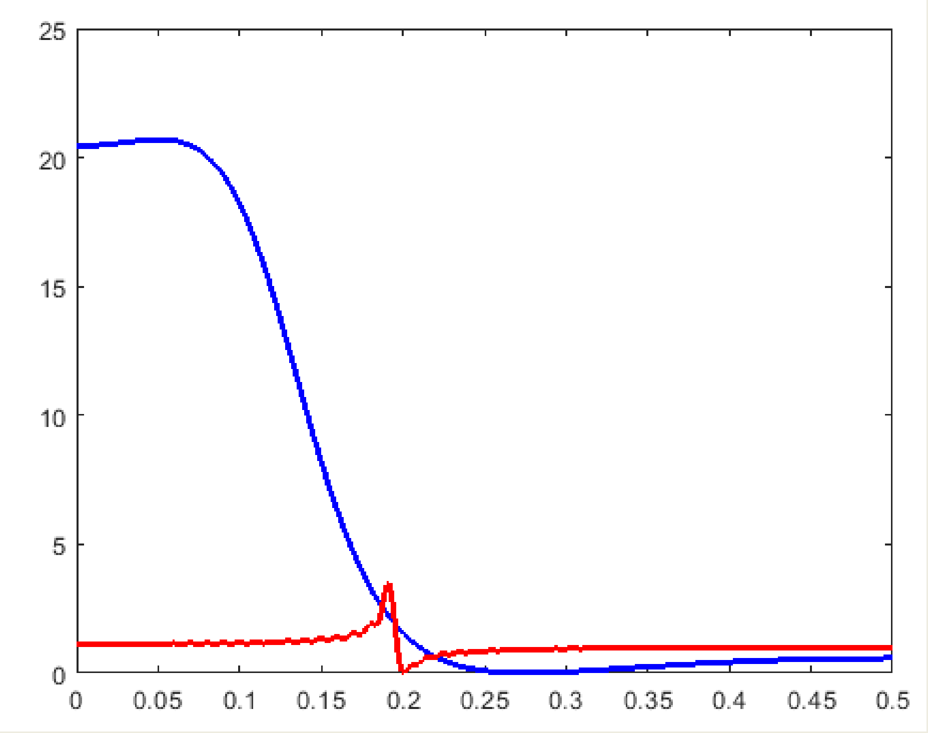
Some implementation aspects

Numerical precision issues



We have to options for zero-pole combination

Magnitude response



f

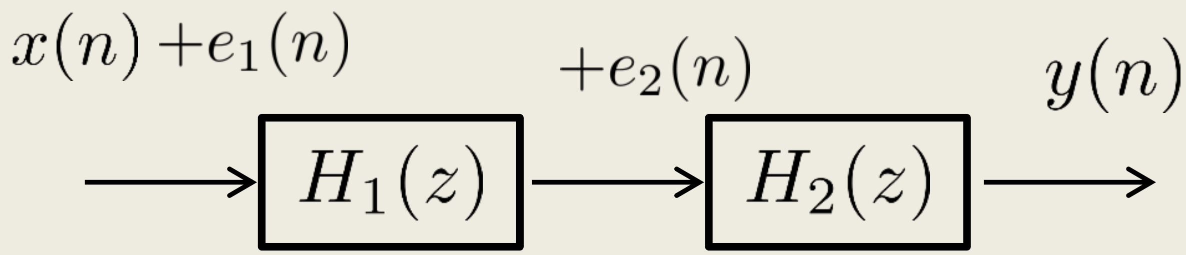
Option 1

We choose option 1. Remains to discuss their order.

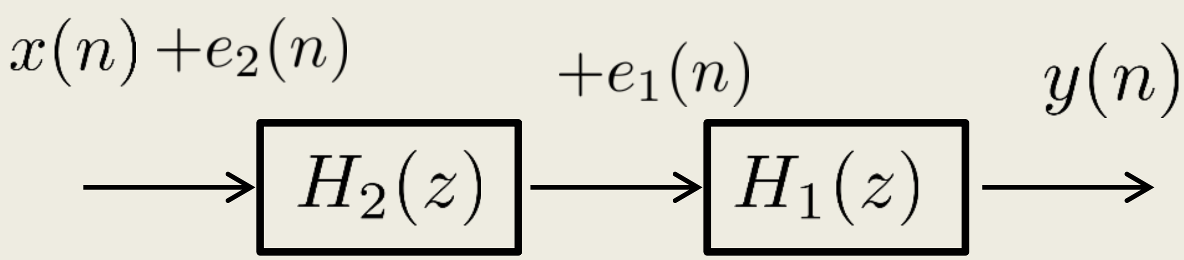
EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues

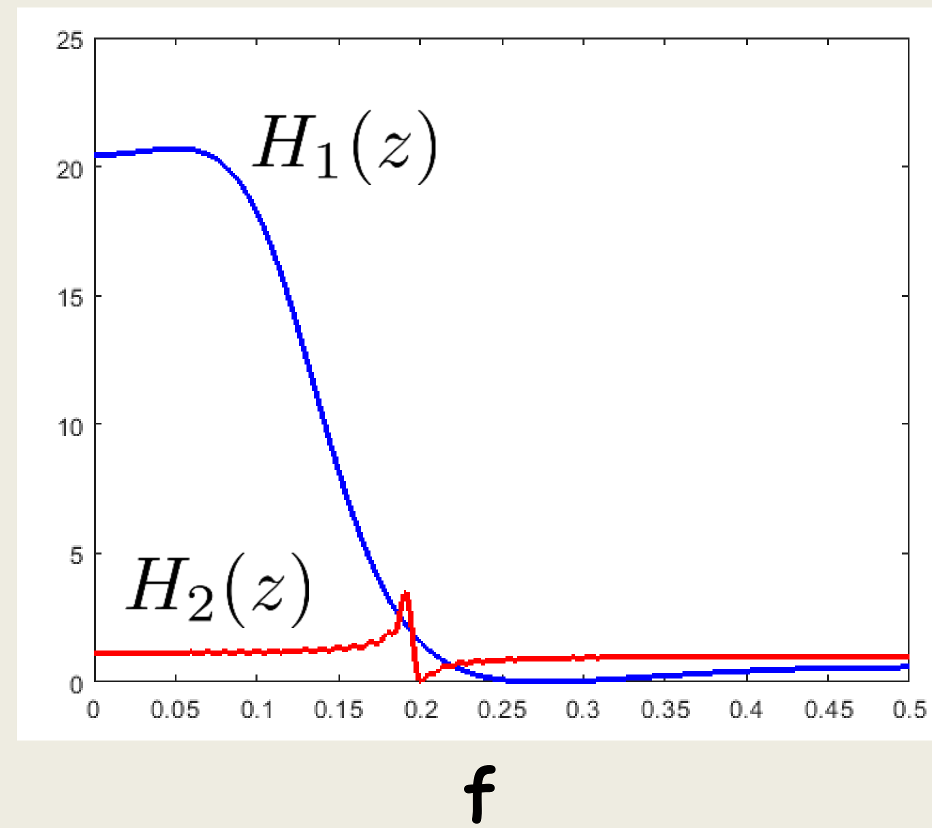


Or



?

Magnitude response



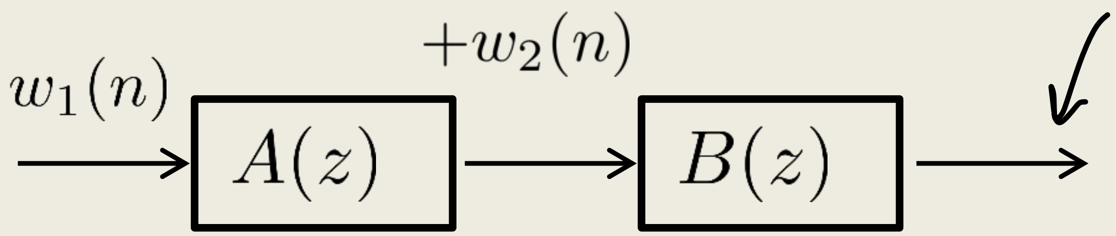
EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues

Study a general model

Noise power ?



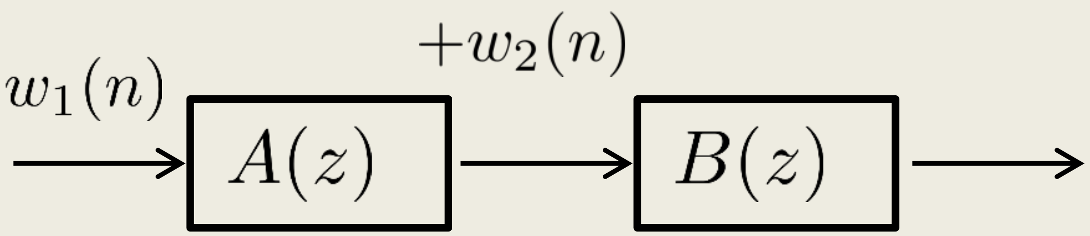
Compute the average output power if the noise sources are unit power random signals

EITF75 Systems and Signals

Some implementation aspects

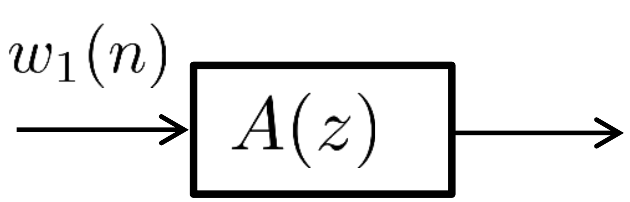
Numerical precision issues

Study a general model



Compute the average output power if the noise sources are **unit** power random signals

"Theorem." Average output power of the below is $\sum_{k=-\infty}^{\infty} |a_k|^2$

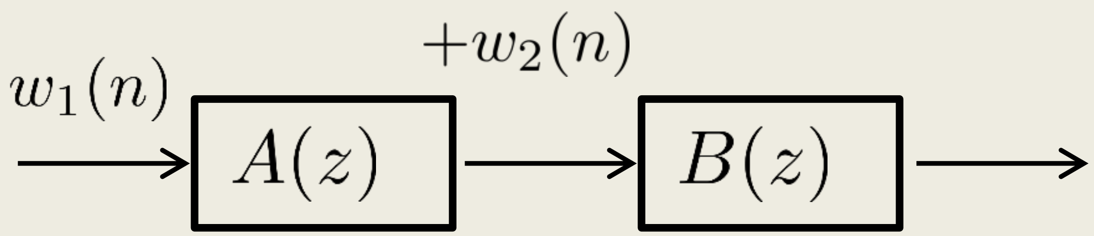


EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues

Study a general model



Output power

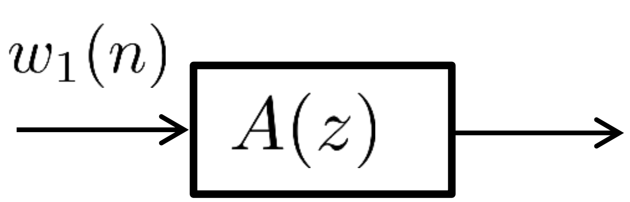
$$\sum_{k=-\infty}^{\infty} |c_k|^2 + \sum_{k=-\infty}^{\infty} |b_k|^2$$

$$c_k = a_k \star b_k$$

$= b_k * a_k$

Compute the average output power if the noise sources are **unit** power random signals

"Theorem." Average output power of the below is $\sum_{k=-\infty}^{\infty} |a_k|^2$

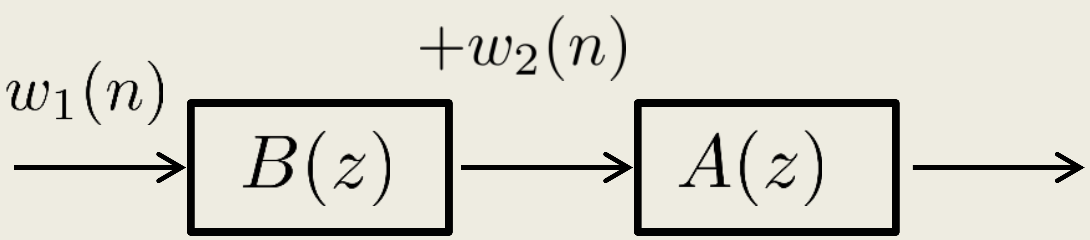


EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues

Study a general model



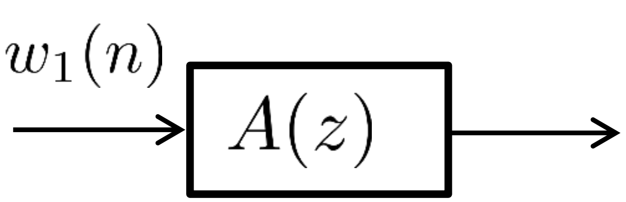
Output power

$$\sum_{k=-\infty}^{\infty} |c_k|^2 + \sum_{k=-\infty}^{\infty} |a_k|^2$$

$$c_k = a_k \star b_k$$

Compute the average output power if the noise sources are unit power random signals

"Theorem." Average output power of the below is $\sum_{k=-\infty}^{\infty} |a_k|^2$



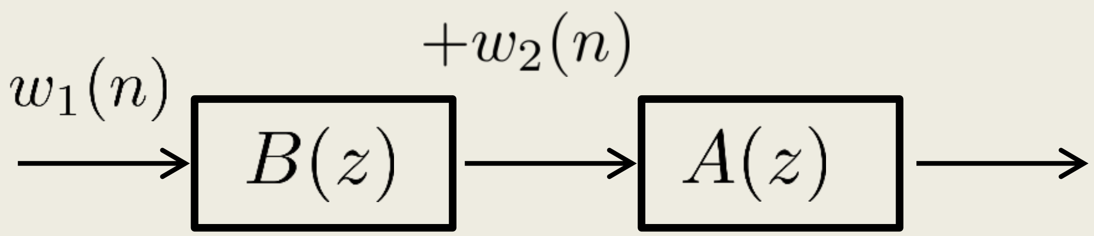
EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues

Study a general model

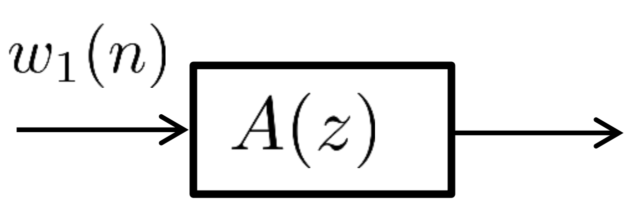
Output power $\qquad c_k = a_k \star b_k$



$$\sum_{k=-\infty}^{\infty} |c_k|^2 + \sum_{k=-\infty}^{\infty} |a_k|^2$$

Place the box with least power last in the chain

"Theorem." Average output power of the below is $\sum_{k=-\infty}^{\infty} |a_k|^2$



EITF75 Systems and Signals

Some implementation aspects

Numerical precision issues

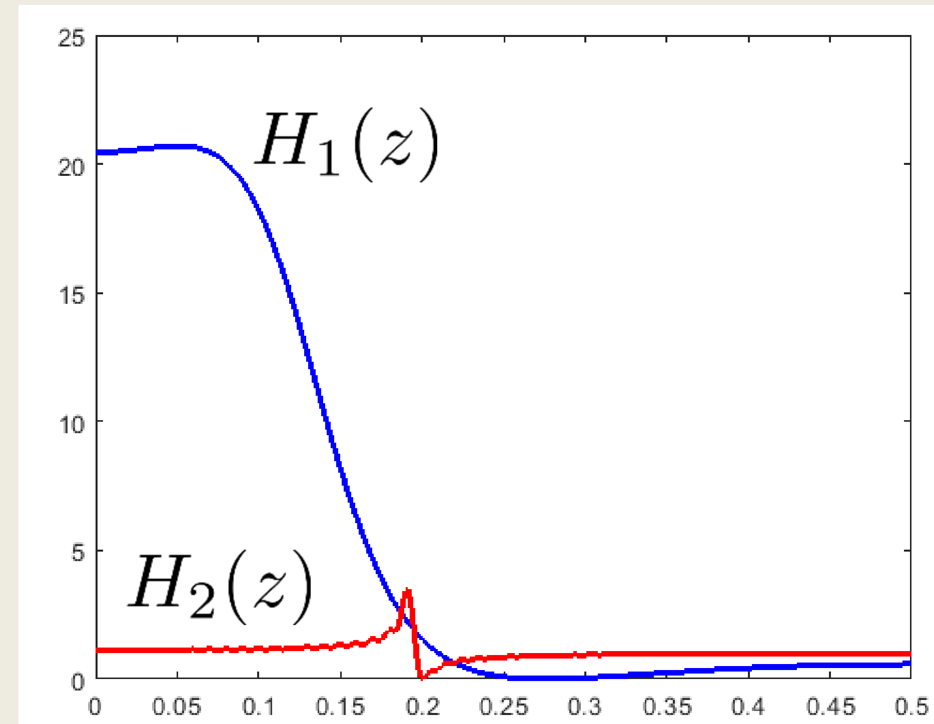
$$\xrightarrow{x(n)+e_1(n)} \boxed{H_1(z)} \xrightarrow{+e_2(n)} \boxed{H_2(z)} \xrightarrow{y(n)}$$

Or

$$\xrightarrow{x(n)+e_2(n)} \boxed{H_2(z)} \xrightarrow{+e_1(n)} \boxed{H_1(z)} \xrightarrow{y(n)}$$

$$\sum_{k=-\infty}^{\infty} |h_1|^2 = \int_{-0.5}^{0.5} |H_1(f)|^2 \mathrm{d}f$$

Magnitude response



f

EITF75 Systems and Signals

Some implementation aspects

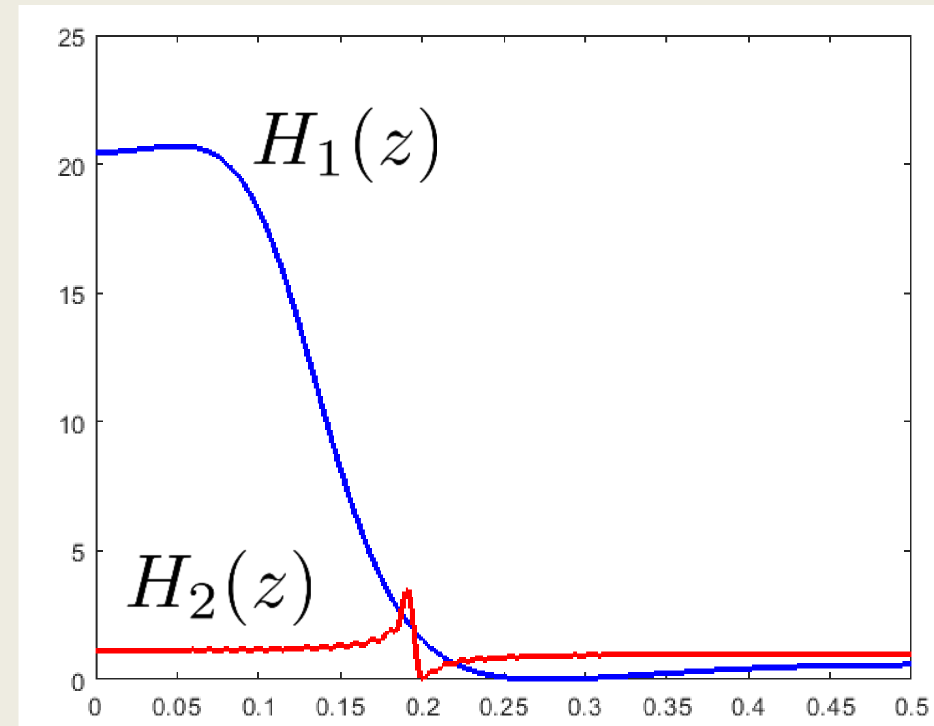
Numerical precision issues

$$\xrightarrow{x(n)+e_1(n)} \boxed{H_1(z)} \xrightarrow{+e_2(n)} \boxed{H_2(z)} \xrightarrow{y(n)}$$

Or

$$\xrightarrow{x(n)+e_2(n)} \boxed{H_2(z)} \xrightarrow{+e_1(n)} \boxed{H_1(z)} \xrightarrow{y(n)}$$

Magnitude response



f

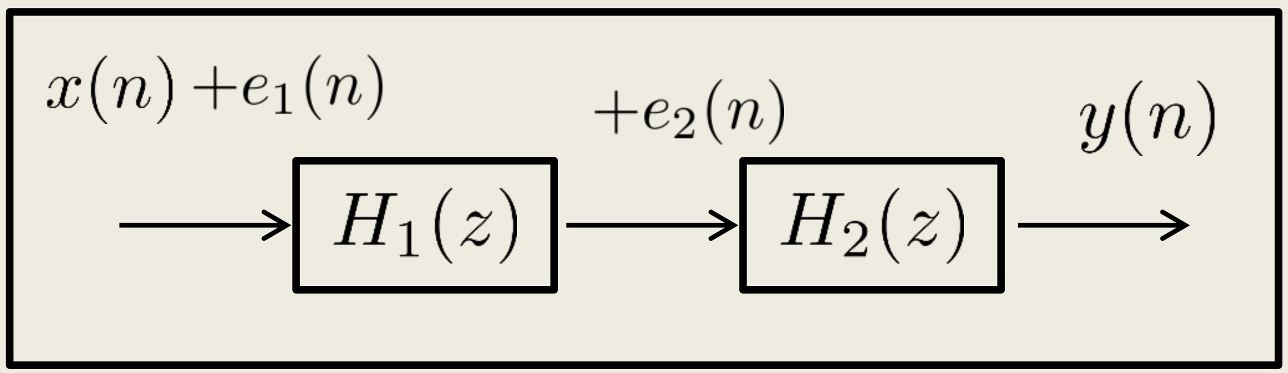
$$\sum_{k=-\infty}^{\infty} |h_1|^2 = \int_{-0.5}^{0.5} |H_1(f)|^2 \mathrm{d}f \gg \int_{-0.5}^{0.5} |H_2(f)|^2 \mathrm{d}f = \sum_{k=-\infty}^{\infty} |h_2|^2$$

EITF75 Systems and Signals

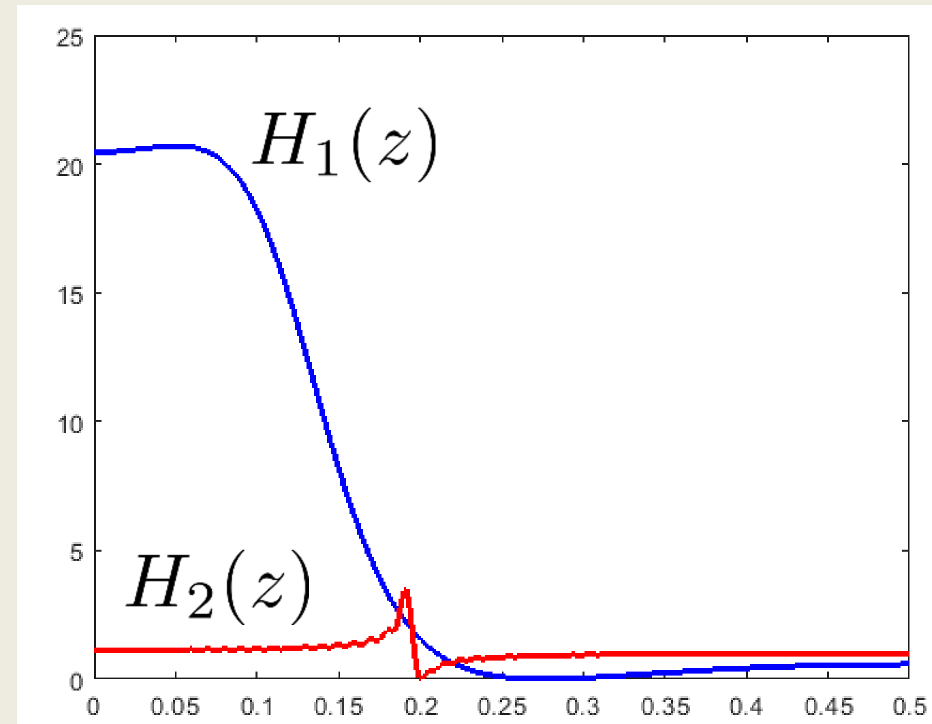
Some implementation aspects

Numerical precision issues

Place $H_2(z)$ last



Magnitude response



f

$$\sum_{k=-\infty}^{\infty} |h_1|^2 = \int_{-0.5}^{0.5} |H_1(f)|^2 \mathrm{d}f \gg \int_{-0.5}^{0.5} |H_2(f)|^2 \mathrm{d}f = \sum_{k=-\infty}^{\infty} |h_2|^2$$