Imperial College London

DSP Lecture 10

Lecture 10

DR TANIA STATHAKI

READER (ASSOCIATE PROFESSOR) IN SIGNAL PROCESSING

IMPERIAL COLLEGE LONDON

FIR and IIR Filters

- The convolution sum description of an LTI discrete-time system can be used to implement the system.
- For an IIR system this approach is not practical as, in that case, the impulse response is of infinite length.
- Here the input-output relation involves a **finite sum of products** as follows:

$$y[n] = -\sum_{k=1}^{N} d_k y[n-k] + \sum_{k=0}^{M} p_k x[n-k]$$

- On the other hand, an FIR system can be implemented, as mentioned above, using the **convolution sum** as follows:

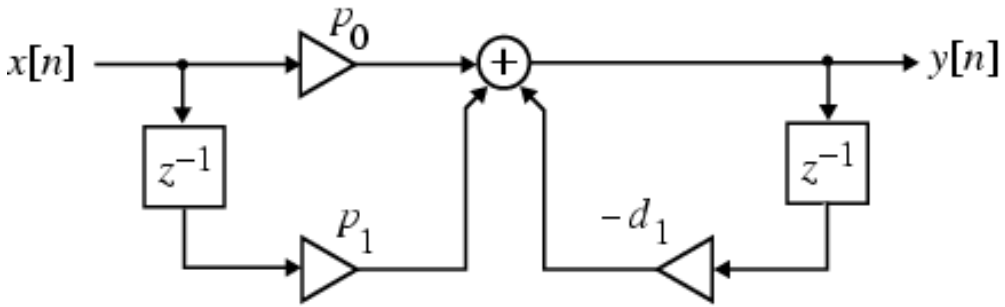
$$y[n] = \sum_{k=0}^{N} h[k]x[n-k]$$

Digital Filter Structures

- The implementation of an LTI digital filter can be either in **software** or **hardware** form, depending on applications.
- In either case, the signal variables and the filter coefficients cannot be represented with finite precision.
- A system (filter) might be implemented using more that one structures.
- It is of practical interest to choose the structure that provides satisfactory performance under finite precision arithmetic.
- In order to design a structure for filter implementation we use basic building blocks (adders, multipliers, delays etc.)

Block Diagram Representation

- For the implementation of an LTI digital filter, the input-output relationship must be described by a valid mathematical relationship.
- Consider the causal first-order LTI digital filter shown below.



- The filter is described by the difference equation

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$$

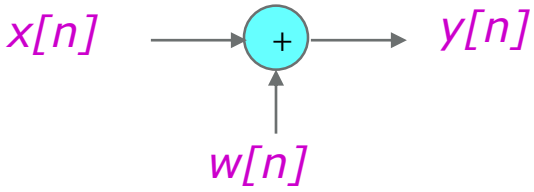
- Using the above equation we can compute $y[n]$ for $n \geq 0$ knowing the initial condition $y[-1]$ and the input $x[n]$ for $n \geq -1$.

$$y[0] = -d_1 y[-1] + p_0 x[0] + p_1 x[-1]$$
$$y[1] = -d_1 y[0] + p_0 x[1] + p_1 x[0]$$
$$y[2] = -d_1 y[1] + p_0 x[2] + p_1 x[1]$$
$$\vdots$$

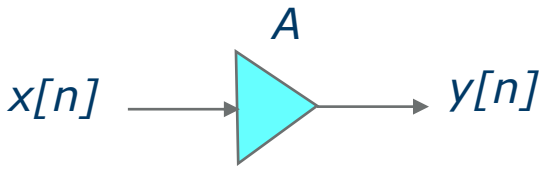
- We can continue this calculation for any value of the time index n.

Block Diagram Representation / Basic Building Blocks

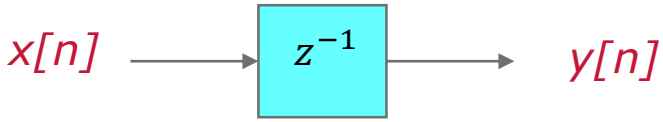
- Each step of the calculation requires knowledge of the previously calculated value of the output sample (delayed value of the output), the present value of the input sample, and the previous value of the input sample (delayed value of the input).
- We say that the first-order difference equation can be interpreted as a valid computational algorithm.
- The computational algorithm of an LTI digital filter can be conveniently represented in a block diagram form using the **basic building blocks** shown below.



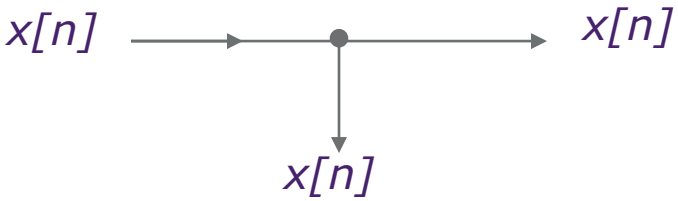
Adder



Multiplier



Unit delay



Pick-off node

Block Diagrams. Advantages.

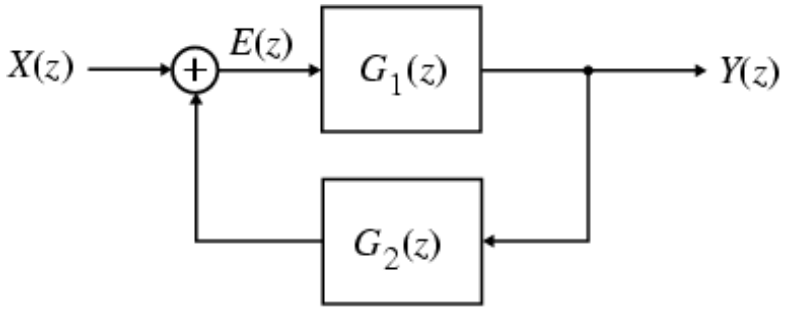
Advantages of block diagram representation:

- Easy to write down the computational algorithm (explicit relation between the output and input) by inspection.
- Easy to manipulate a block diagram to derive other “equivalent” block diagrams yielding different computational algorithms.
- Easy to determine the hardware requirements.

Analysis of Block Diagrams. Example 1.

- Consider the single-loop feedback structure shown below. The output $E(z)$ of the adder is $E(z) = X(z) + G_2(z)Y(z)$. But from the figure we see that:

$$Y(z) = G_1(z)E(z)$$



- Eliminating $E(z)$ from the previous two equations we arrive at

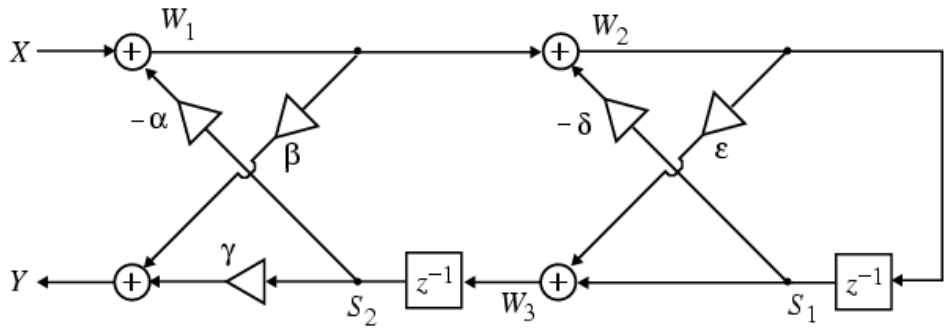
$$[1 - G_1(z)G_2(z)]Y(z) = G_1(z)X(z)$$

which leads to

$$H(z) = \frac{Y(z)}{X(z)} = \frac{G_1(z)}{1 - G_1(z)G_2(z)}$$

Analysis of Block Diagrams. Example 2.

- Analyze the cascaded lattice structure shown below where the z −dependence of signal variables are not shown for brevity.



- The output signals of the four adders are given by

$$W_1 = X - \alpha S_2$$
$$W_2 = W_1 - \delta S_1$$
$$W_3 = S_1 + \varepsilon W_2$$
$$Y = \beta W_1 + \gamma S_2$$

- From the figure we observe that:

$$S_2 = z^{-1} W_3$$
$$S_1 = z^{-1} W_2$$

Analysis of Block Diagrams. Example 2 cont.

- Substituting the last two relations in the first four equations we get:

$$W_1 = X - \alpha z^{-1} W_3$$
$$W_2 = W_1 - \delta z^{-1} W_2$$
$$W_3 = z^{-1} W_2 + \varepsilon W_2$$
$$Y = \beta W_1 + \gamma z^{-1} W_3$$

- From the second equation we get $W_2 = W_1/(1 + \delta z^{-1})$ and from the third equation we get $W_3 = (\varepsilon + z^{-1})W_2$.
- Combining the last two equations we get

$$W_3 = \frac{\varepsilon + z^{-1}}{1 + \delta z^{-1}} W_1$$

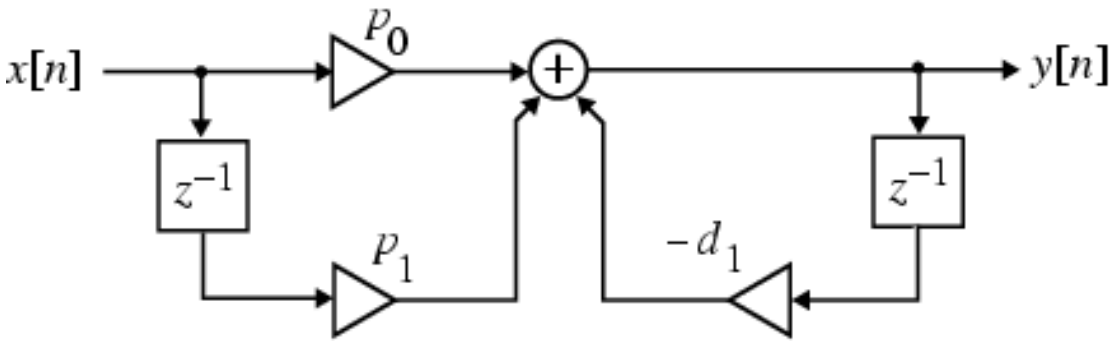
- Substituting the above equation in $W_1 = X - \alpha z^{-1} W_3$, $Y = \beta W_1 + \gamma z^{-1} W_3$ we finally arrive at

$$H(z) = \frac{Y}{X} = \frac{\beta + (\beta\delta + \gamma\varepsilon) z^{-1} + \gamma z^{-2}}{1 + (\delta + \alpha\varepsilon) z^{-1} + \alpha z^{-2}}$$

Canonic and Non-canonic Structures

- A digital filter structure is said to be **canonic** if the number of delays in the block diagram representation is equal to the order of the transfer function.
- Otherwise, it is a **non-canonic** structure.
- The structure shown below is non-canonic as it employs two delays to realize a first-order difference equation.

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$$



Equivalent Structures

- Under infinite precision arithmetic any given realization of a digital filter behaves identically to any other equivalent structure.
- However, in practice, due to the finite wordlength limitations, a specific realization behaves totally differently from its other equivalent realizations.
- Hence, it is important to choose a structure that has the least quantization effects when implemented using finite precision arithmetic.
- One way to arrive at such a structure is to determine a large number of equivalent structures, analyze the finite wordlength effects in each case, and select the one showing the least effects.
- In certain cases, it is possible to develop a structure that by construction has the least quantization effects.

Basic/Direct Form of FIR Digital Filter Structures

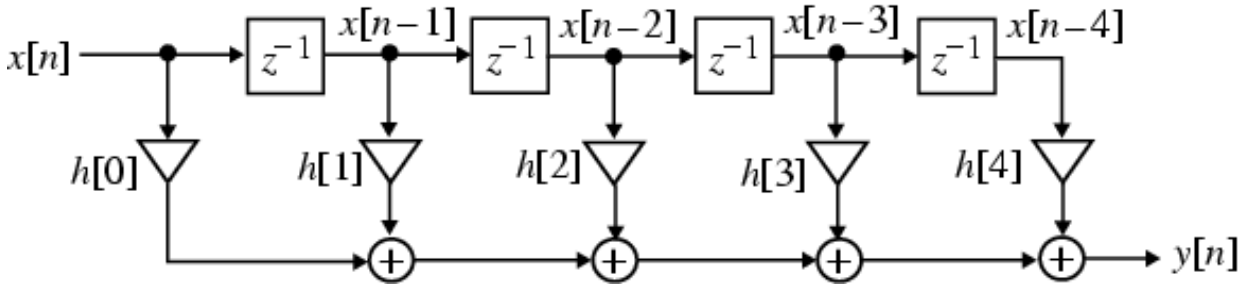
- A causal FIR filter of order N is characterized by a transfer function $H(z)$ given by

$$H(z) = \sum_{n=0}^{N} h[n]z^{-n}$$

- In the time-domain the input-output relation of the above FIR filter is given by

$$y[n] = \sum_{k=0}^{N} h[k]x[n-k]$$

- An FIR filter of order N is characterized by $N+1$ coefficients and, in general, requires $N+1$ multipliers and N two-input adders.
- Structures in which the multiplier coefficients are precisely the coefficients of the transfer function are called **direct form** structures.
- A direct form realization of an FIR filter can be readily developed from the convolution sum description as indicated below for $N = 4$.



Equivalent Structures

- Two digital filter structures are defined to be **equivalent** if they have the same transfer function.
- There is a number of methods for the generation of equivalent structures.
- However, a fairly simple way to generate an equivalent structure from a given realization is via the **transpose operation**.

Transpose Operation

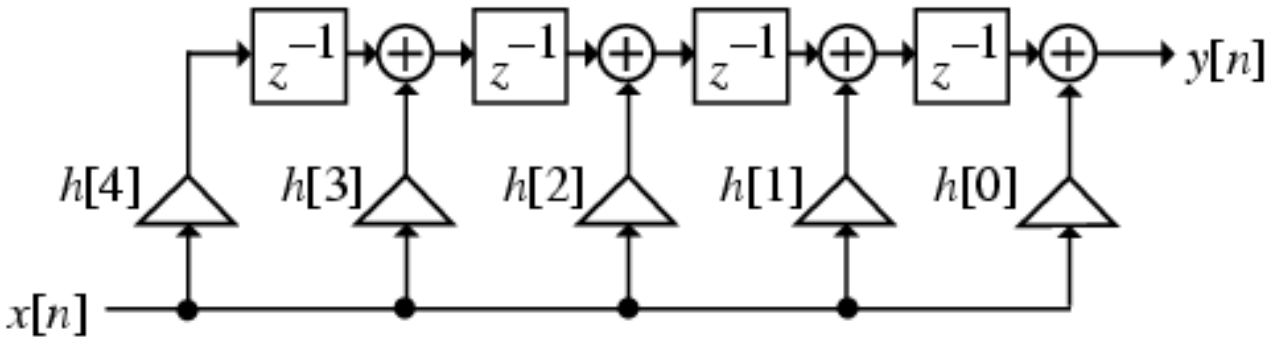
1. Reverse all paths.
2. Replace pick-off nodes by adders, and vice versa.
3. Interchange the input and output nodes.

All other methods for developing equivalent structures are based on a specific algorithm for each structure.

- There are literally an infinite number of equivalent structures realizing the same transfer function.
- It is thus impossible to develop all equivalent realizations.

Direct Form of FIR Digital Filter Structures

- An analysis of this structure yields
  $y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + h[3]x[n-3] + h[4]x[n-4]$
  which is precisely of the form of the convolution sum description
- The direct form structure shown on the previous slide is also known as a **tapped delay line** or a **transversal filter**.
- The transpose of the direct form structure shown earlier is indicated below.
- Both direct form structures are canonic with respect to delays.



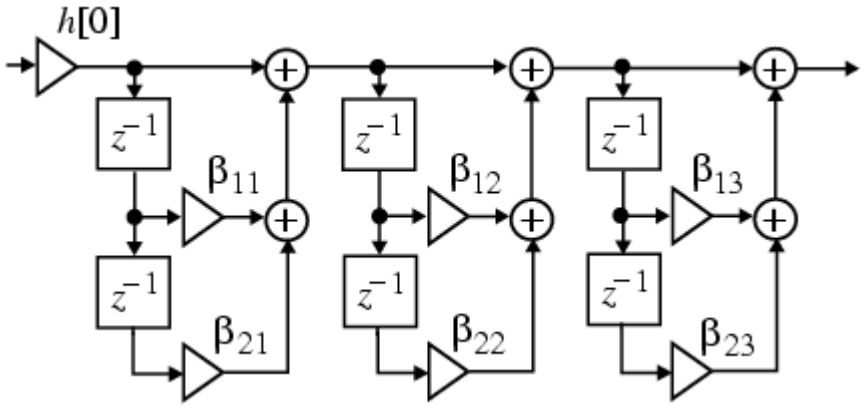
Cascade Form FIR Digital Filter Structures

- A higher-order FIR transfer function can also be realized as a cascade of second-order FIR sections and possibly a first-order section.
- To this end we express $H(z)$ as

$$H(z) = h[0]\prod_{k=1}^{K}(1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2})$$

where $K = \frac{N}{2}$ if N is even, and $K = \frac{N+1}{2}$ if N is odd, with $\beta_{2K} = 0$.

- A cascade realization for $N = 6$ is shown below.



- Each second-order section in the above structure can also be realized in the transposed direct form.