EE402 - Discrete Time Systems **Spring 2018**

Lecture 4

Lecturer: Asst. Prof. M. Mert Ankarali

Realization of Discrete Time Systems / Filters / Controllers

This lecture will cover some fundamental block-diagram realization techniques for discrete-time systems, filters, & controllers represented by Z-Domain transfer functions or difference equations.

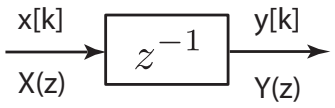
Block-diagram realizations are extremely useful practically to implement the system/filter/controller on a physical embedded platform. Whether your goal is to programming the discrete filter/controller on a microcontroller or embedding the structure on a FPGA module, a block diagram representation is always helpful.

In the last phase of the course, we will also actively use block-diagram representation to obtain a state-space realization from a given discrete-time system.

The most fundamental block for a discrete-time system is the unit delay operator

$$y[k] = x[k-1]$$
$$Y(z) = z^{-1}X(z)$$

which is represented by the following block-diagram



In this lecture, our goal is to realize different kinds of discrete-time transfer functions using this fundamental block as the main brick.

Realization of FIR Systems

A third order (order is fixed for the sake of clarity) FIR (Finite impulse response) discrete time system has the following difference equation and transfer function

$$y[k] = b_0 x[k] + b_1 x[k-1] + b_2 x[k-2] + b_3 x[k-3]$$
$$Y(z) = \left(b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}\right) X(z)$$

From inspection it is easy to see that we need at least three unit delay blocks (and memory elements) to construct a full realization. Below the block-diagram realization of a third order FIR is given in Fig. 4.1

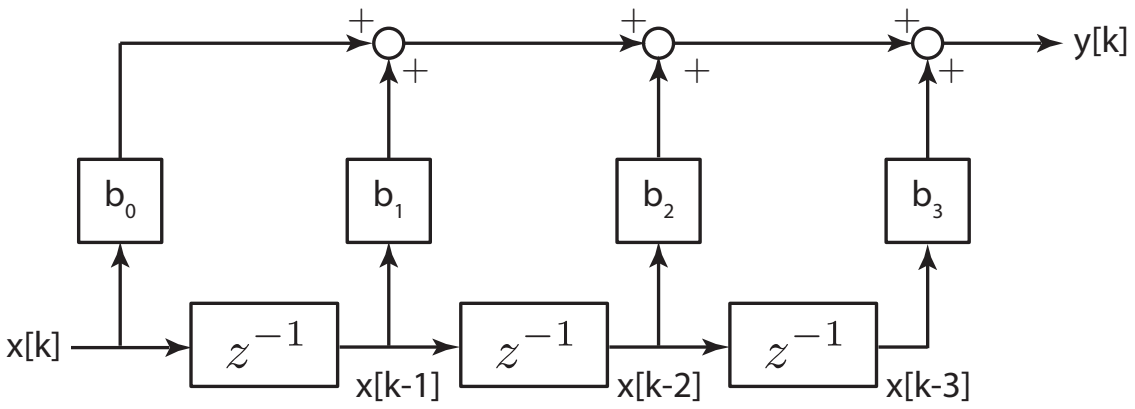


Figure 4.1: Block diagram realization of a third order FIR system

Realization of IIR Systems with Static Numerator Dynamics

In this part we will analyze a special case if IIR (Infinite impulse response) systems where the input dynamics is static, i.e. there is no direct delayed term of the input in the difference equation. Again for the sake of clarity let's assume that the discrete system is third order. For such a system, the difference equation and the transfer function can be written as

$$y[k] = -a_1 y[k-1] - a_2 y[k-2] - a_3 y[k-3] + b_0 x[k]$$
$$Y(z) = \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} X(z)$$

Similar to the FIR case we also need minimum three delay blocks to realize this system, However, now delay blocks are in the feedback-loop. The block digram representation of the given IIR system is given in Fig. 4.2.

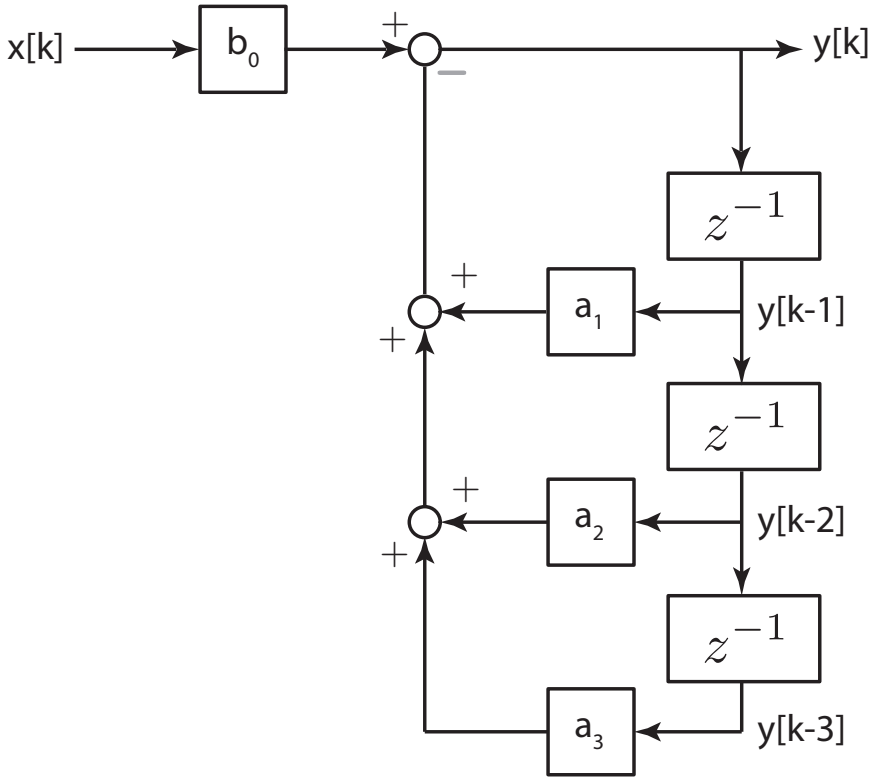


Figure 4.2: Block diagram realization of a third order IIR system with static numerator/input dynamics

Realization of General Discrete-Time Systems

Now we will attempt to obtain a realization of more general discrete-time systems/filters/controllers. Again for the sake oc clarity let's keep the order (maximum order of z^{-1} in the equation) of the system to 3. A general third order discrete time system can be expressed with following difference equation and transfer function

$$y[k] = -a_1 y[k-1] - a_2 y[k-2] - a_3 y[k-3] + b_0 x[k] + b_1 x[k-1] + b_2 x[k-2] + b_3 x[k-3]$$
$$Y(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} X(z)$$

Non-minimal Realization / Direct Programming

One way of realizing a discrete-time system is simply combining the block diagrams of special cases (i.e. FIR and IIR with static input dynamics) given in Figures 4.1 & 4.2. The block diagram realization obtained with this method can be observed in Fig. 4.3

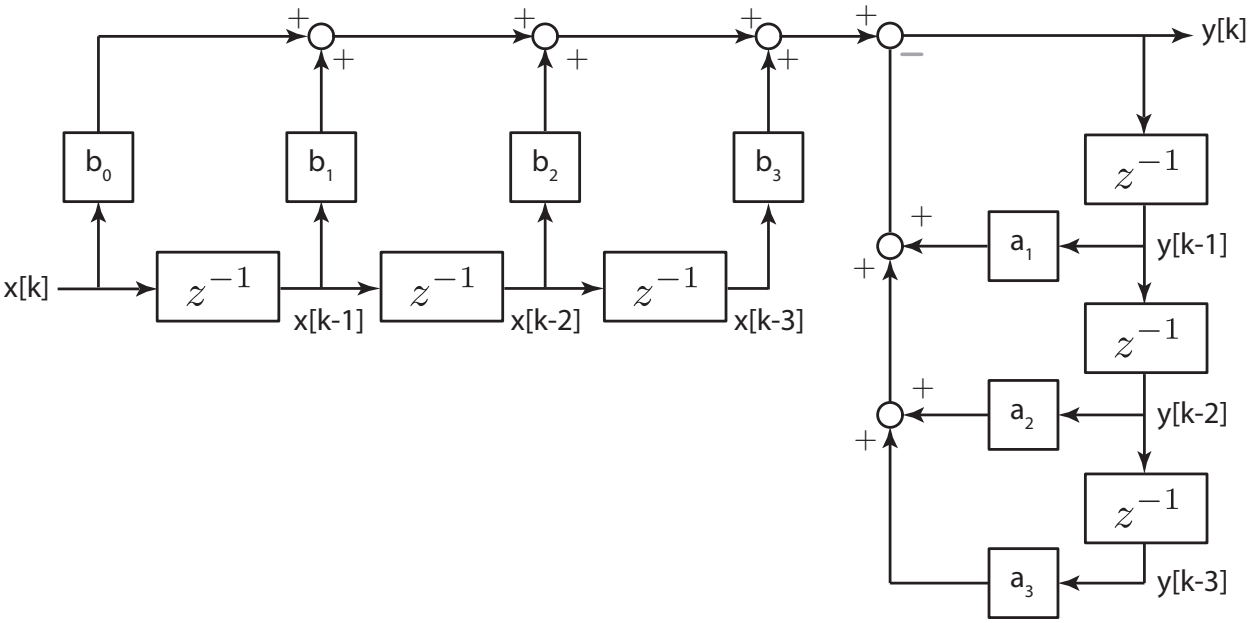


Figure 4.3: Direct/non-minimal realization of a third order discrete time dynamical system

The obvious problem in this realization is that even though the transfer function represents a third order discrete-dynamical system, the "order" of the block diagram is indeed 6 not 3. Because there exist 6 different memory blocks, i.e. delay elements. If we for example obtain a state-space model from this block diagram we would obtain a system with 6 states.

Canonical Realization I / Standard Programming

In this method of realization, we will use the fact the system is LTI. Let's consider the transfer function of the system and let's perform some LTI operations.

$$\begin{aligned}
Y(z) &= \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} X(z) \\
&= \left(b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}\right) \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} X(z) \\
&= G_2(z) G_1(z) X(z) \text{ where} \\
G_1(z) &= \frac{H(z)}{X(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \\
G_2(z) &= \frac{Y(z)}{H(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}
\end{aligned}$$

As you can see we introduced an intermediate variable $h[k]$ with a Z-transform of $H(z)$, First transfer function, which is an IIR system with static input dynamics operates on $x[n]$ and produces an output. Second transfer function operates on $h[n]$ and produces output $x[n]$. If we write the difference equations of both systems we obtain

$$\begin{aligned}
h[k] &= -a_1 h[y-1] - a_2 h[k-2] - a_3 h[k-3] + x[k] \\
y[k] &= b_0 x[k] + b_1 h[k-1] + b_2 h[k-2] + h_3 x[k-3]
\end{aligned}$$

As it can be seen that the delay/shifting operations are only performed on the signal $h[k]$ and maximum delay operation is by 3 samples. Basically if we utilize this structure we can draw a minimal block diagram representation as given in Fig. 4.4. If we obtain a state-space model from this block diagram, the form will be in *controllable canonical form*. We will cover this later in the semester. Thus we can call this representation also as *controllable canonical realization*.

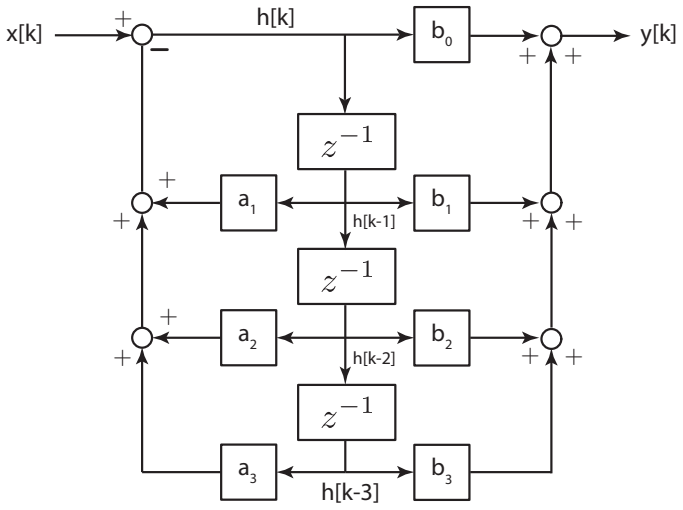


Figure 4.4: A minimal block diagram realization of a discrete time system obtained with standard programming (Canonical representation I)

Canonical Realization II

In this method will obtain a different minimal realization. The process will be different and this the block diagram will have different topology. Let's start with the transfer function and perform some grouping based on the delay elements.

$$Y(z)(1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}) = (b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3})X(z)$$
$$Y(z) = b_0 X(z) + z^{-1}\left(b_1 X(z) - a_1 Y(z)\right) + z^{-2}\left(b_2 X(z) - a_2 Y(z)\right) + z^{-3}\left(b_3 X(z) - a_3 Y(z)\right)$$
$$Y(z) = b_0 X(z) + z^{-1}\left\{\left(b_1 X(z) - a_1 Y(z)\right) + z^{-1}\left[\left(b_2 X(z) - a_2 Y(z)\right) + z^{-1}\left(b_3 X(z) - a_3 Y(z)\right)\right]\right\}$$

As you can see we have only z^{-1} terms in the representation there is a special toplogy embedded inside the expression. If we convert it to the block diagram form we obtain the structure given in Fig. 4.5. If we obtain a state-space model from this block diagram, the form will be in *observable canonical form*. Thus we can call this representation also as *observable canonical realization*. This form and representation is the dual of the previous representation.

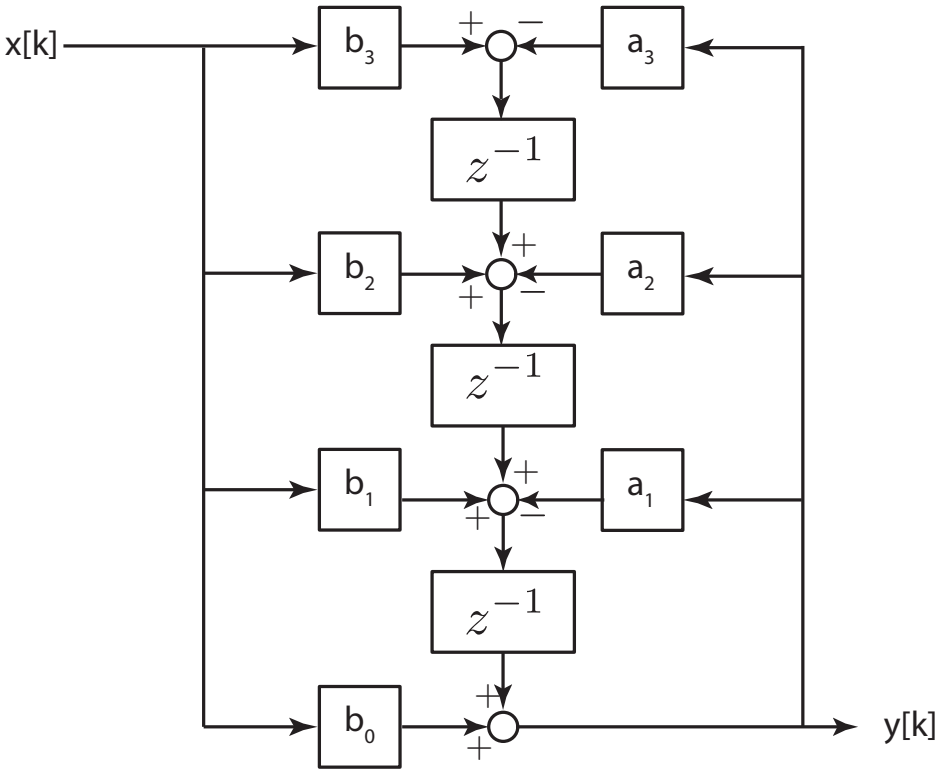


Figure 4.5: A minimal block diagram realization of the 3^{rd} order discrete time system obtained with Canonical representation II