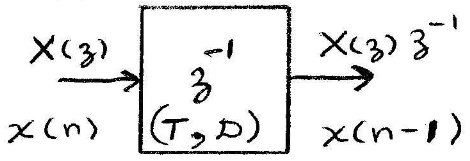
Lesson 11 - Digital Filter Realization Forms

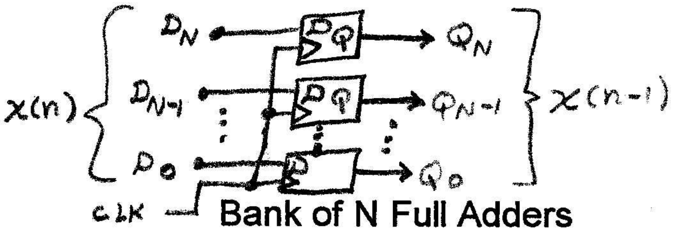
I. Basic Operations

Once an H(z) has been determined, it can be implemented using the following discrete system "building blocks", which may physically exist, or be emulated in software.

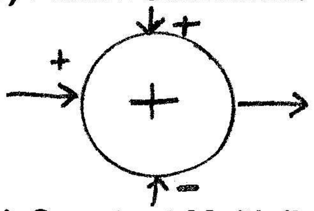
(a) Unit Delay Block



Bank of N D FFs

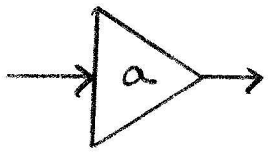


(b) Adder/Subtractor



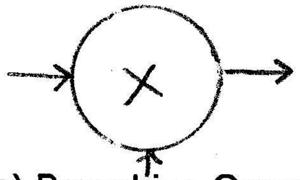
Bank of N Full Adders

(c) Constant Multiplier



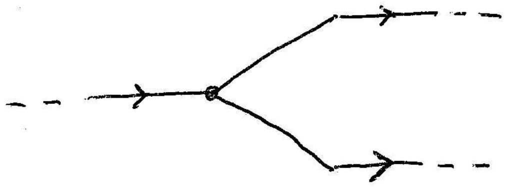
Parallel Multiplier

(d) SIgnal Multiplier



Parallel Multiplier

(e) Branching Operation



Y-connected Wires

II. Direct Form

A. Direct Form #1

First put $H(z)$ in "negative exponent" form $\Rightarrow$ divide top and bottom by z^n

Example: $$H(z)=\frac{3+3.6\cdot z^{-1}+0.6\cdot z^{-2}}{1+0.1\cdot z^{-1}-0.2\cdot z^{-2}}$$

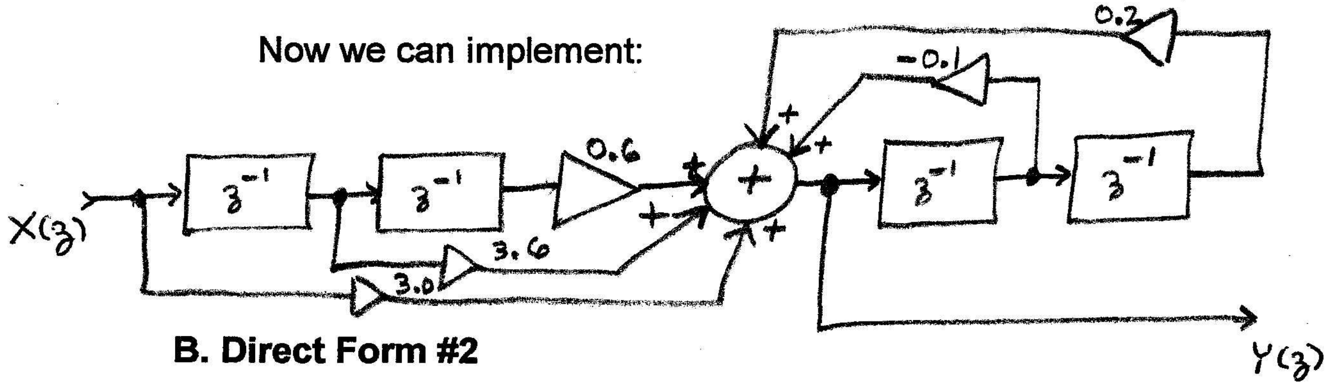
$$\frac{Y(z)}{X(z)}=\frac{3+3.6\cdot z^{-1}+0.6\cdot z^{-2}}{1+0.1\cdot z^{-1}-0.2\cdot z^{-2}}$$

Cross-multiply

$$Y(z)\cdot\left(1+0.1\cdot z^{-1}-0.2\cdot z^{-2}\right)=X(z)\cdot\left(3+3.6\cdot z^{-1}+0.6\cdot z^{-2}\right)$$

$$Y(z)=X(z)\cdot\left(3+3.6\cdot z^{-1}+0.6\cdot z^{-2}\right)-Y(z)\cdot\left(0.1\cdot z^{-1}-0.2\cdot z^{-2}\right)$$

Now we can implement:



B. Direct Form #2

Outline of approach: *Think of H(z) as a ratio of two polynomials in z, where the numerator of H(z) is the polynomial N(z), and the denominator of H(z) is the polynomial D(z).*

$$H(z)=\frac{Y(z)}{X(z)}=\frac{N(z)}{D(z)}$$

$$Y(z)=H(z)\cdot X(z)=\frac{N(z)\cdot X(z)}{D(z)}=N(z)\cdot W(z)$$

Where the auxilliary function is defined as $W(z)\triangleq\dfrac{X(z)}{D(z)}$

Example:

$$H(z)=\frac{Y(z)}{X(z)}=\frac{3+3.6\cdot z^{-1}+0.6\cdot z^{-2}}{1+0.1\cdot z^{-1}-0.2\cdot z^{-2}}$$

$$Y(z)=\left(3+3.6\cdot z^{-1}+0.6\cdot z^{-2}\right)\cdot\frac{X(z)}{\left(1+0.1\cdot z^{-1}-0.2\cdot z^{-2}\right)}$$

$$Y(z)=\left(3+3.6\cdot z^{-1}+0.6\cdot z^{-2}\right)\cdot W(z) \qquad (11.1)$$

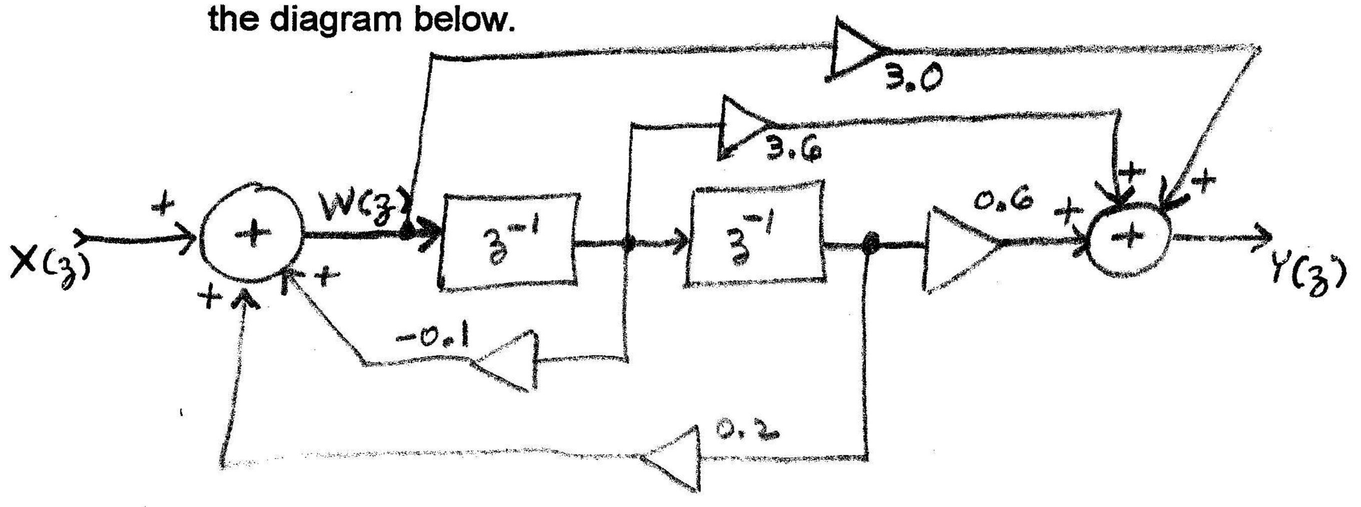
$$W(z)=\frac{X(z)}{\left(1+0.1\cdot z^{-1}-0.2\cdot z^{-2}\right)} \qquad (11.2)$$

$$X(z)=W(z)\cdot\left(1+0.1\cdot z^{-1}-0.2\cdot z^{-2}\right) \qquad (11.3)$$

$$W(z)=X(z)-0.1\cdot z^{-1}\cdot W(z)+0.2\cdot z^{-2}\cdot W(z) \qquad (11.4)$$

Now we can use the "*Bootstrap Technique*" of analog computer programming: We start by drawing a summing block (at the left side of the paper) and assume that we have W(z) at the output of this summing block. Next, we make it become a "self-fulfilling prophecy". We make it happen by drawing a chain of delay blocks to the right of the summing block, and feed appropriately scaled outputs of the delay chain back around to the input of the summing block, as dictated by Eqn (11.4).
Once W(z) has been realized, we can use Eqn (11.1) to realize the output function (which is what we really care about implementing).

Note that Y(z) is formed at the output of the right summing block in the diagram below.



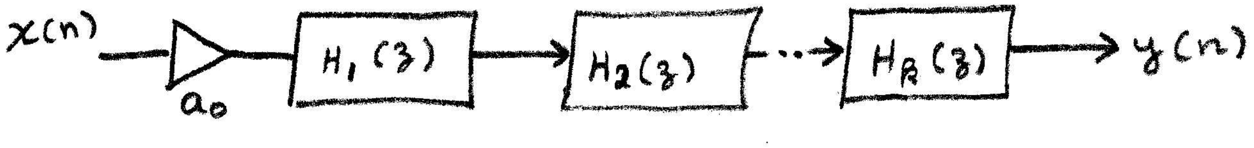
Compare Direct Form #1 with Direct Form #2. Note the savings in the required number of delay blocks in Direct Form #2! Thus Direct Form #2 is usually preferred over Direct Form #1. In a hardwired implementation of this discrete system, this savings translates into fewer D Flip Flops needed in the design. In a DSP chip (software) implementation of this system, this savings translates into fewer memory locations needed.

Note that the H(z) denominator coefficients end up appearing in feedback paths in the implementation, while H(z) numerator coefficients end up appearing in feed-forward paths.

The above implementation represents a standard (Direct Form #2) 2nd order digital filter topology. Often this second-order structure is used as a building block for higher-order filters. The reason we don't go beyond 2nd-order systems with this approach is due to increasing coefficient accuracy requirements as the order is increased. (Coefficient sensitivity increases with the system order.)

C. Cascade Form

$$H(z)=a_0\cdot H_1(z)\cdot H_2(z)\cdot H_3(z)\cdot\ldots\cdot H_k(z)=a_0\cdot\prod_{i=1}^{k} H_i(z)$$



Example:

$$H(z)=\frac{Y(z)}{X(z)}=\frac{3+3.6\cdot z^{-1}+0.6\cdot z^{-2}}{1+0.1\cdot z^{-1}-0.2\cdot z^{-2}}$$

$$=\frac{3\cdot z^2+3.6\cdot z+0.6}{z^2+0.1\cdot z-0.2}=\frac{3\cdot(z+1)\cdot(z+0.2)}{(z+0.5)\cdot(z-0.4)}$$

Thus we may realize H(z) as a cascade of two first-order sections:

$$H(z)=3\cdot\frac{z+1}{z+0.5}\cdot\frac{z+0.2}{z-0.4}=a_0\cdot H_1(z)\cdot H_2(z)$$

It doesn't matter which pairing of numerators and denominators we choose, although different coefficient sensitivity may result.

Now we may implement each 1st-order section in Direct Form #2:

Implementing the first 1st-order section:

$$H_1(z)=\frac{Y(z)}{X(z)}=\frac{1+z^{-1}}{1+0.5\cdot z^{-1}}$$

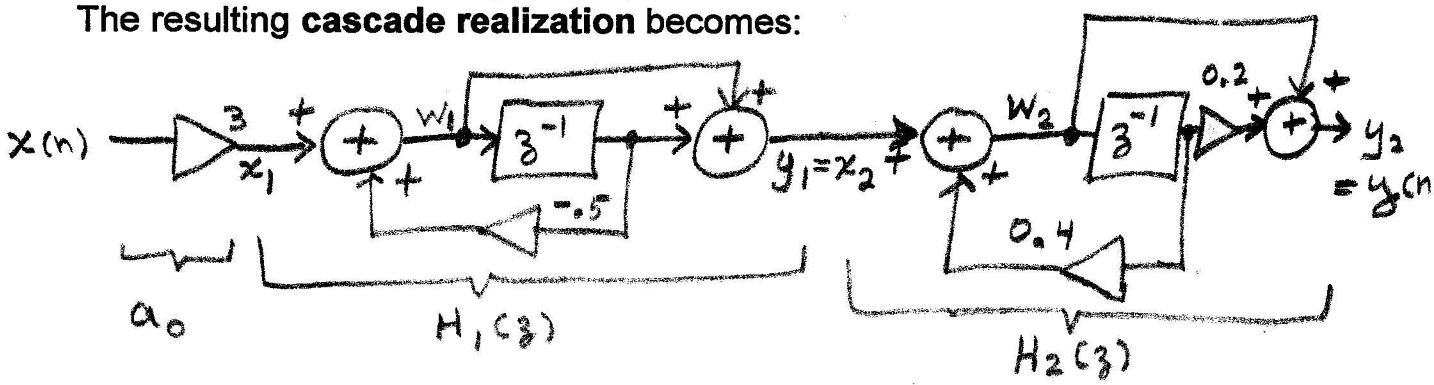
$$Y(z)=\left(1+z^{-1}\right)\cdot\underbrace{\left[\frac{X(z)}{\left(1+0.5\cdot z^{-1}\right)}\right]}_{=W(z)}=W(z)+z^{-1}\cdot W(z)$$

$$W(z)=\frac{X(z)}{1+0.5\cdot z^{-1}}$$

$$W(z)+0.5\cdot z^{-1}\cdot W(z)=X(z)$$

$$W(z)=X(z)-0.5\cdot z^{-1}\cdot W(z)$$

The second 1st-order stage, is implemented in identical fashion.
The resulting **cascade realization** becomes:



Usually, 2nd-order stages are cascaded, since often complex-conjugate root pairs are present, which cannot be factored into less than a 2nd-order equation without going to complex coefficients.

D. Parallel Form (based upon PFE)

$$H(z) = A + H_1(z) + H_2(z) + \ldots + H_n(z)$$

$$H(z) = A + \sum_{i=1}^{n} H_i(z)$$

Example

$$H(z) = 3\cdot\frac{z+1}{z+0.5}\cdot\frac{z+0.2}{z-0.4}$$

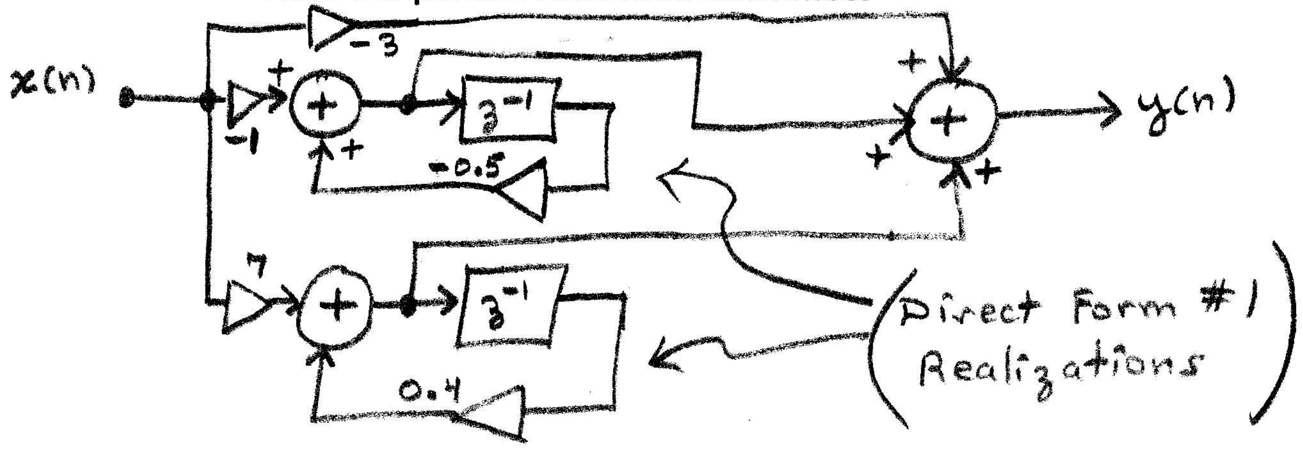
$$\frac{H(z)}{z} = \frac{3}{z}\cdot\frac{z+1}{z+0.5}\cdot\frac{z+0.2}{z-0.4} = \frac{-3}{z} + \frac{-1}{z+0.5} + \frac{7}{z-0.4}$$

$$H(z) = -3 + \frac{-z}{z+0.5} + \frac{7\cdot z}{z-0.4}$$

$$H(z) = a_0 + H_1(z) + H_2(z)$$

Where $a_0 = -3$, $H_1(z) = \dfrac{-1}{1+0.5\cdot z^{-1}}$, and $H_2(z) = \dfrac{7}{1-0.4\cdot z^{-1}}$

Thus the parallel realization becomes:



In general, 2nd-order blocks will be used in parallel realization, since sometimes complex conjugate roots will exist, in which case we will not be able to break down further than 2nd-order form, as we combine complex-conjugate root pairs.