ECE 410 **DIGITAL SIGNAL PROCESSING** **D. Munson**
University of Illinois **Chapter 7** **A. Singer**

Solution of Difference Equations (D.E.'s) Using z-Transform

Just as the Laplace transform was used to aid in the solution of linear differential equations, the z-transform can be used to aid in the solution of linear difference equations. Recall that linear, constant coefficient differential equations could be converted into algebraic equations by transforming the signals in the equation using the Laplace transform. Derivatives could be mapped into functions of the Laplace transform variable *s*, through the derivative law for Laplace transforms. Similarly, delayed versions of a sequence can be mapped into algebraic functions of *z*, using one of the delay rules for z-transforms.

In the case of continuous-time linear systems described by differential equations, in order to find the response of such a linear system to an particular input, the differential equations needed to be solved, using either time-domain or Laplace transform methods. For an N^{th}-order differential equation, in general N conditions on the output were needed in order to specify the output in response to a given input. Similarly, for linear difference equations of N^{th}-order, N pieces of information are needed to find the output for a given input. Unlike the continuous-time case, difference equations can often be simply iterated forward in time if these N conditions are consecutive. That is, given $y[-N+1]$, ... $y[-1]$, then re-writing

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

in the form

$$y[0] = -\sum_{k=1}^{N} a_k x[n-k] + \sum_{k=0}^{N} b_k x[n-k]$$

from which $y[0]$ could be found. Iterating this process forward could find each value of the output without ever explicitly obtaining a general expression for $y[n]$.

In this chapter we will explore the z-transform for the explicit solution of linear constant coefficient difference equations. The properties of the z-transform that we have developed can be used to map the difference equations describing the relationship between the input and the output, into a simple set of linear algebraic equations involving the z-transforms of the input and output sequences. By solving the resulting algebraic equations for the z-transform of the output, we can then use the methods we've developed for inverting the transform to obtain an explicit expression for the output. We begin with an example.

Example

We revisit this simple linear, homogeneous difference equation, now using the unilateral z-transform. Again consider the difference equation

$$y[n] - 3y[n-1] = 0, n \geq 0, y[-1] = 2$$

Taking unilateral z-transform of both sides, and using the delay property, we obtain

$$Y(z) - 3z^{-1}\left[Y(z) + z\,y[-1]\right] = 0$$
$$Y(z)\left[1 - 3z^{-1}\right] = 6$$

which can be solved for $Y(z)$, yielding

$$Y(z) = \frac{6z}{z-3}$$
$$y[n] = 6(3)^n u[n]$$

Another, slightly more involved example repeats a previous problem as well.

Example

Consider the following homogenous, linear constant coefficient difference equation, defined for nonnegative n and with initial conditions shown

$$y[n] + 4y[n-1] + 4y[n-2] = 0, \quad n \geq 0, \quad y[-1] = y[-2] = 1$$

Taking the z-transform of both sides, again using the delay property and including the initial conditions, we obtain

$$Y(z) + 4z^{-1}\left[Y(z) + zy[-1]\right] + 4z^{-2}\left[Y(z) + z\,y[-1] + z^2\,y[-2]\right] = 0$$
$$Y(z)[1 + 4z^{-1} + 4z^{-2}] = -4y[-1] - 4z^{-1}y[-1] - 4y[-2]$$
$$Y(z) = \frac{-8 - 4z^{-1}}{1 + 4z^{-1} + 4z^{-2}}$$
$$= \frac{-8z^2 - 4z}{z^2 + 4z + 4}.$$

Since this is not in strictly proper rational form, we expand $z^{-1}Y(z)$ in a partial fraction expansion, yielding

$$\frac{Y(z)}{z} = \frac{-8z-4}{(z+2)^2} = \frac{A_1}{z+2} + \frac{A_2}{(z+2)^2}.$$

Since we have repeated roots, we first seek the coefficient of the highest order root, A_2. By cross multiplying, we obtain

$$-8z - 4 = A_1(z+2) + A_2.$$

From this equation, we can actually identify both coefficients. First by selecting $z = -2$, we obtain

$$A_2 = (-8z-4)\Big|_{z=-2} = 12.$$

We can also immediately identify A_1 by matching terms of z on both sides, since the constant terms on each side yield

$$-8z = A_1 z$$
$$-8 = A_1.$$

Putting these terms into the PFE, we obtain

$$Y(z) = \frac{-8z}{(z+2)} + \frac{12z}{(z+2)^2}.$$

We can now use linearity of the z-transform to invert each of the terms separately, obtaining,

$$y[n] = -8(-2)^n - 6n(-2)^n, \; n \geq 0.$$

We now consider a case where the difference equation contains an input, or drive term, such that we no longer have a homogenous difference equation.

Example

Consider the following linear constant coefficient difference equation.

$$y[n+2]-\frac{3}{2}y[n+1]+\frac{1}{2}y[n]=\left(\frac{1}{3}\right)^n u[n],\;\; y[0]=4, y[1]=0$$

Taking the unilateral z-transform of both sides and using the advance property, we obtain

$$z^2\left[Y(z)-y[0]-z^{-1}y[1]\right]-\frac{3}{2}z\left[Y(z)-y[0]\right]+\frac{1}{2}Y(z)=\frac{z}{z-\frac{1}{3}}$$

$$z^2\left[Y(z)-4\right]-\frac{3}{2}z\left[Y(z)-4\right]+\frac{1}{2}Y(z)=\frac{z}{z-\frac{1}{3}}$$

$$Y(z)\left[z^2-\frac{3}{2}z+\frac{1}{2}\right]=\frac{z}{z-\frac{1}{3}}+4z^2-6z$$

We can now solve for $Y(z)$ and keep the terms on the right hand side separated into two groups, namely,

$$Y(z)=\frac{1}{\left[z^2-\frac{3}{2}z+\frac{1}{2}\right]}\left[\underbrace{\frac{z}{z-\frac{1}{3}}}_{\substack{\text{term due}\\ \text{to input}}}+\underbrace{4z^2-6z}_{\substack{\text{term due}\\ \text{to initial}\\ \text{conditions}}}\right].$$

We can now write the z-transform as a sum of two terms, one due to the input, and one due to the initial conditions. Recall from our analysis of linear constant coefficient difference equations that these correspond to the zero-state response and the zero-input response of the system. Taking these two terms separately, again through linearity of the transform, we have that

$$Y(z)=T_1(z)+T_2(z)$$

where

$$T_1(z)=\frac{z}{\left(z^2-\frac{3}{2}z+\frac{1}{2}\right)\left(z-\frac{1}{3}\right)}$$

$$T_2(z)=\frac{4z^2-6z}{z^2-\frac{3}{2}z+\frac{1}{2}}$$

Here, $T_1(z)$ is the z-transform of the zero-state response, and $T_2(z)$ is the z-transform of the zero-input response. We can then take a partial fraction expansion of each of the terms independently. For the first term, we find it convenient to express the partial fraction expansion as

$$\frac{T_1(z)}{z} = \frac{1}{\left(z-\frac{1}{2}\right)(z-1)\left(z-\frac{1}{3}\right)} = \frac{A_1}{\left(z-\frac{1}{2}\right)} + \frac{A_2}{(z-1)} + \frac{A_3}{\left(z-\frac{1}{3}\right)}.$$

This leads to

$$A_1 = -12, A_2 = 3, A_3 = 9$$

$$\frac{T_1(z)}{z} = \frac{-12}{\left(z-\frac{1}{2}\right)} + \frac{3}{(z-1)} + \frac{9}{\left(z-\frac{1}{3}\right)},$$

and the resulting zero-state response is given by

$$y_x[n] = -12\left(\frac{1}{2}\right)^n + 3 + 9\left(\frac{1}{3}\right)^n, n \geq 0.$$

For the zero-input response term, we have that

$$\frac{T_2(z)}{z} = \frac{4z-6}{\left(z-\frac{1}{2}\right)(z-1)} = \frac{B_1}{\left(z-\frac{1}{2}\right)} + \frac{B_2}{(z-1)},$$

from which we can quickly solve for the constants, yielding

$$B_1 = 8, \; B_2 = -4,$$

which gives the PFE for the zero-input response as

$$T_2(z) = \frac{8z}{z-\frac{1}{2}} - \frac{4z}{z-1}.$$

So, zero-input response is

$$y_s[n] = 8\left(\frac{1}{2}\right)^n - 4u, \; n \geq 0.$$

Putting the zero-state response and the zero-input response together, we obtain the total response as

$$y[n] = y_x[n] + y_s[n] = -4\left(\frac{1}{2}\right)^n - 1 + 9\left(\frac{1}{3}\right)^n, \quad n \geq 0.$$

In general, this method of solution can be applied to linear constant coefficient difference equations of arbitrary order. Note that while in this particular case we applied the time-advance property of the unilateral z-transform, when solving a difference equation of the form

$$y[n] + a_1 y[n-1] + \cdots + a_N y[n-N] = x[n], \quad n \geq 0,$$

with initial conditions $y[-k]$, k =1, ... N, use the Delay Property #2.

General Form of Solution of Linear Constant Coefficient Difference Equations (LCCDE)s

In this section, we will derive the general form of a solution to a linear constant coefficient difference equation. We will prove that the zero-state response (response to the input, when state is initially zero) is given by a *convolution*. Consider the following difference equation

$$y[n+K]+a_1 y[n+K-1]+\cdots+a_N y[n]=x[n],\quad n\geq 0$$

together with initial conditions $y[k]$, $k = 0, 1, \ldots, K-1$. Taking the one-sided z-transform of both sides, and using the Advance Property, we obtain

$$z^K\left[Y(z)-\sum_{m=0}^{K-1} y[m]z^{-m}\right]+a_1 z^{K-1}\left[Y(z)-\sum_{m=0}^{K-2} y[m]z^{-m}\right]+\cdots+a_{K-1}z\left[Y(z)-y[0]\right]+a_K Y(z)=X(z).$$

By defining

$$S(z)=z^K\sum_{m=0}^{K-1} y[m]z^{-m}+a_1 z^{K-1}\sum_{m=0}^{K-2} y[m]z^{-m}+\cdots+a_{K-1}zy[0],$$

we have that

$$Y(z)[z^K+a_1 z^{K-1}+\cdots+a_K]=X(z)+S(z),$$

where the *characteristic polynomial* is given by

$$z^K+a_1 z^{K-1}+\cdots+a_K\,.$$

We now define the *transfer function* $H(z)$,

$$H(z)=\frac{1}{z^K+a_1 z^{K-1}+\ldots+a_K},$$

we obtain that

$$Y(z)=H(z)[\underbrace{X(z)}_{\substack{\text{term due}\\ \text{to the input}}}+\underbrace{S(z)}_{\substack{\text{term due to}\\ \text{initial conditions}}}].$$

Notice that the decomposition property holds with

$$\begin{cases} y_s[n]=Z^{-1}\{H(z)\ S(z)\} \\ y_x[n]=Z^{-1}\{H(z)\ X(z)\} \end{cases}.$$

Both homogeneity and superposition hold with respect to y_S and y_X because the z-transform is linear. Linear constant coefficient difference equations (LCCDE)s describe linear systems, which we have already explored in the time-domain (sequence-domain). It is worthwhile to consider the form of the solution that $y_s[n]$ will take.

Consider first the case when the roots of the characteristic polynomial are distinct. In this case, we have

$$\frac{S(z)H(z)}{z}=\frac{B_1}{z-r_1}+\frac{B_2}{z-r_2}+\cdots+\frac{B_K}{z-r_K}.$$

From the definition of $S(z)$, z is a factor in $S(z)$, so there is no need for a B_0/z term in the partial fraction expansion. Multiplying by z, we have

$$S(z)H(z)=\frac{B_1 z}{z-r_1}+\frac{B_2 z}{z-r_2}+\cdots+\frac{B_K z}{z-r_K},$$

from which we can easily recover the sequence

$$y_s[n] = \sum_{i=1}^{K} B_i\,(r_i)^n, \quad n \geq 0,$$

which is in the same form as $y_H[n]$ that we have already seen in the sequence-domain solution of LCCDEs.

We can now observe the form of $y_X[n]$. Since we have that

$$y_x[n] = Z^{-1}\{H(z)X(z)\},$$

the partial fraction expansion shows that $y_x[n]$ will involve terms <u>both</u> in $y[n]$ and in $x[n]$. We can also rewrite $y_X[n]$ using the convolution property:

$$y_x[n] = \sum_{m=0}^{n} h[m]\,x[n-m],$$

Equation 10-1

where,

$$\begin{aligned} h[m] = Z^{-1}\{H(z)\} &= Z^{-1}\left\{z\frac{H(z)}{z}\right\} \\ &= Z^{-1}\left\{z\left(\frac{D_0}{z} + \frac{D_1}{z-r_1} + \ldots + \frac{D_K}{z-r_K}\right)\right\} \\ &= \begin{cases} D_0 + \sum_{i=1}^{K} D_i\left(r_i\right)^n & n = 0 \\ \sum_{i=1}^{K} D_i\left(r_i\right)^n & n \geq 1 \end{cases} \end{aligned}$$

which is in the same form as $y_H[n]$ and $y_S[n]$ for n ≥ 1. So, we see that $y_X[n]$ is given by a convolution of the input with $h[n] = Z^{-1}\,\{H(z)\}$. What is $\{h[n]\}_{n=0}^{\infty}$? This can be interpreted as the system unit pulse response (u.p.r.), or impulse response, assuming zero initial conditions.

<u>Definition</u>
The unit-pulse sequence, or the *discrete-time impulse,* is given by

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}.$$

The system response to a unit pulse, or *impulse*, is given by,

$$y[n] = y_x[n] \quad = \sum_{m=0}^{n} h[m]\delta[n-m] = h[n].$$

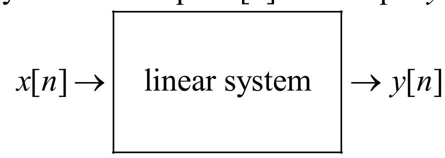
↓

assuming zero
initial conditions

We can explore the use of the impulse response to derive the response to more general signals through another example.

Example

Consider the following linear system with input $x[n]$ and output $y[n]$ as shown



Suppose that when the input $x[n] = \delta[n]$ with zero initial conditions, then the output satisfies $y[n]=a^n$ for $n \geq 0$. Again, assuming zero initial conditions (i.e. the system is *initially at rest*), find $y[n]$ due to the input $x[n] = b^n$, $n \geq 0$.

Solution:

Given $h[n] = a^n$, $n \geq 0$, we know that the output satisfies $y[n] = y_X[n]$, since the initial conditions are all zero, i.e. the system is initially at rest. From **Equation 10-1Equation 10-1**, we have that

$$y[n] = \sum_{m=0}^{n} a^m b^{n-m} = b^n \sum_{m=0}^{n} \left(\frac{a}{b}\right)^m$$

$$= \frac{b^{n+1}}{b} \frac{1-\left(\frac{a}{b}\right)^{n+1}}{1-\frac{a}{b}} \qquad (a \neq b)$$

$$= \frac{b^{n+1} - a^{n+1}}{b-a}$$

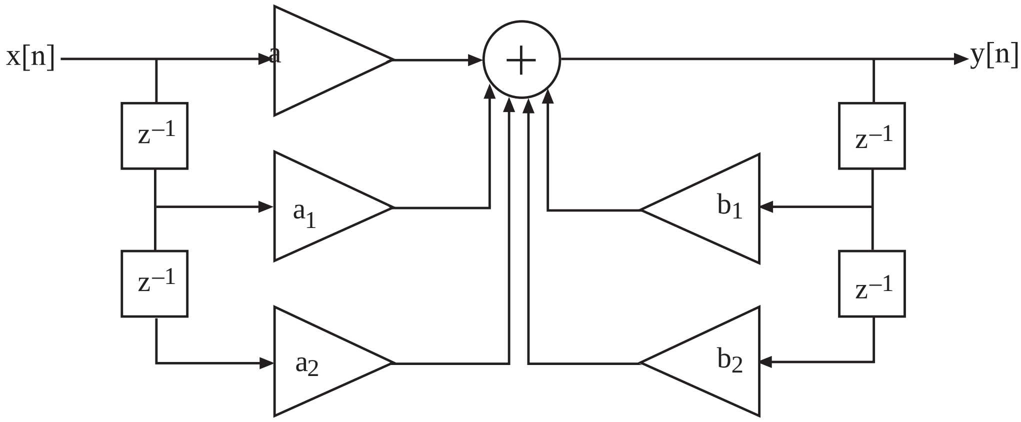
Comments

This discussion and these examples lead us to a number of conclusions about the solutions to linear constant coefficient difference equations. First, we can show (and we will see in the next sections) that the solution to a linear constant coefficient difference equation will have a essentially the same form when the input is merely shifted in time. Also, we will see that a similar for is maintained for inputs that are linear combinations of shifted versions of the input. For example, the response to an input of the form $x[n]$ will be similar in form to the response to the input $x[n] - 2\,x[n-1]$. We will also see that the solution methods developed here, as well as the unilateral z-transform, can be modified to accommodate situations when the input is applied earlier or later than for $n = 0$. While we discussed situations here that included both the zero-input response and the zero-state response, in practice we are generally interested in the zero-state response, or equivalently, we are interested in the response to an input when the system of interest is *initially at rest.* The reason for this is that we either have a system where the initial conditions are all zero, or for a stable system, such that the roots of the characteristic polynomial are all of modulus less than unity, $|r_i| < 1$, and that after some time, $y_S[n]$ has sufficiently decayed, such that for time scales of interest for a given application, $y[n] \approx y_X[n]$. As a result, from this point forward, we will assume that systems under discussion are initially at rest, and that all initial conditions are set to zero. As a result, the output of a linear system will be taken as the zero-state response, and we will be interested in the convolution relationship between the input and the output.

System Block Diagrams

Common 2^{nd}-Order Digital Filter Structures

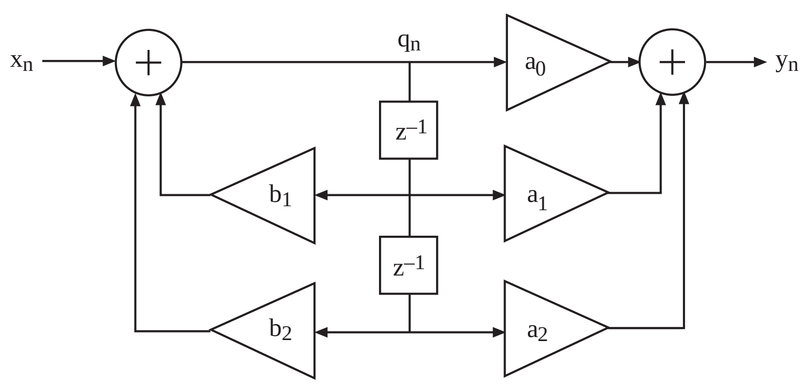
Direct Form 1:



Students: Write D. E. and take z-transform of both sides to show

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}}$$

Direct Form 2:



Has same transfer function as Direct Form 1. Let's show this.
Hard to write y_n in terms of x_n. Introduce "dummy variable" q_n. Write two D.E.'s.

1) $y_n = a_0\, q_n + a_1\, q_{n-1} + a_2\, q_{n-2}$

$$\Rightarrow\ Y(z) = a_0\, Q(z) + a_1\, z^{-1}\, Q(z) + a_2\, z^{-2}\, Q(z)$$

2) $q_n = x_n + b_1 q_{n-1} + b_2 q_{n-2}$

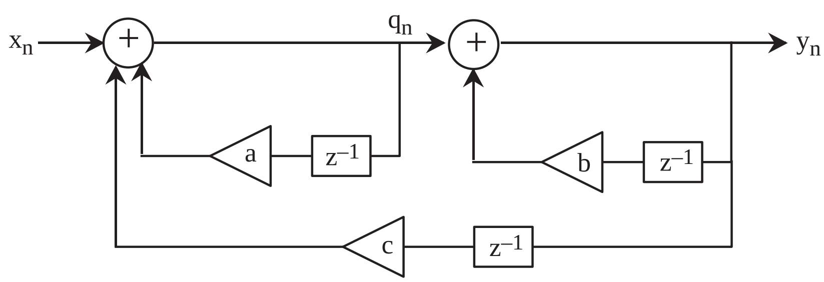
$$\Rightarrow Q(z)\left[1 - b_1 z^{-1} - b_2 z^{-2}\right] = X(z)$$

$$\Rightarrow Y(z) = \left[a_0 + a_1 z^{-1} + a_2 z^{-2}\right]\frac{X(z)}{1 - b_1 z^{-1} - b_2 z^{-2}}$$

$$\Rightarrow H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}} \quad ✓$$

Example

Find H(z) for



Can write D.E.'s and then z-transform them, or else just write z-transforms directly:

$$Y(z) = Q(z) + b\, z^{-1}\, Y(z) \qquad (1)$$

$$Q(z) = X(z) + a\, z^{-1}\, Q(z) + c\, z^{-1}\, Y(z) \qquad (2)$$

$$(1) \Rightarrow Q(z) = Y(z)\left[1 - b\, z^{-1}\right]$$

Substitute into (2):

$$Y(z)\left[1 - b\, z^{-1}\right] = X(z) + a\, z^{-1}\, Y(z)\left[1 - b\, z^{-1}\right] + c\, z^{-1}\, Y(z)$$

$$\Rightarrow Y(z)\left[1 - b\, z^{-1} - a\, z^{-1} + ab\, z^{-2} - c z^{-1}\right] = X(z)$$

$$\Rightarrow \boxed{H(z) = \frac{1}{1 - (a + b + c) z^{-1} + ab z^{-2}}}$$

Notes on digital filter implementation

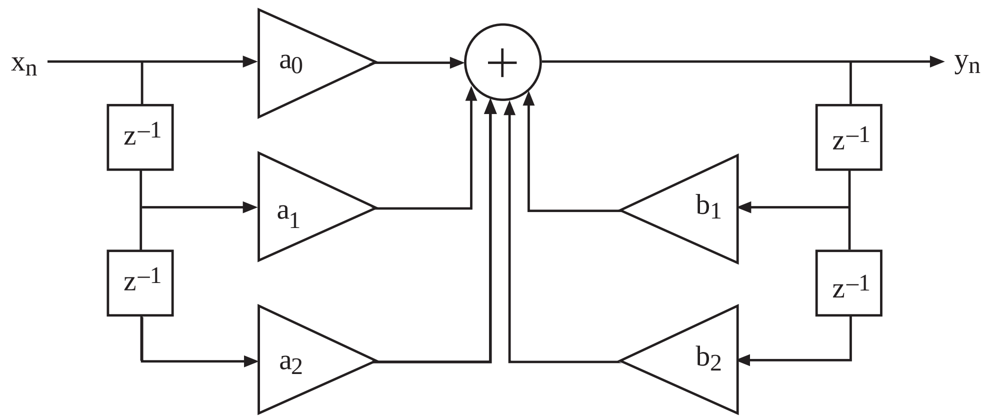
$\{h_n\}$ and H(z) are <u>input-output</u> descriptions of digital filters. Given an input $\{x_n\}$, we can use either $\{h_n\}$ or H(z) to determine the output $\{y_n\}$. In this sense, both $\{h_n\}$ and H(z) <u>summarize</u>

the behavior of the system. However, neither $\{h_n\}$ nor H(z) tell us what the internal structure of the digital filter looks like. Indeed, for any given H(z), there are an infinite number of filter structures that will all have this same transfer function. For a second-order transfer function

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}}$$

the Direct Form 1 and Direct Form 2 structures are just the two most obvious possibilities.

At this point, you may wonder how the filter structure or diagram relates to the actual filter implementation. The answer is multifaceted. Let's consider the Direct Form I structure as an example.



Suppose we implement this filter in a DSP microprocessor. Then, the first thing we must realize is that the system is clocked. The clock is not shown in our digital filter diagram. Ordinarily it takes many clock cycles, corresponding to many microprocessor instructions, to compute a single value of the output sequence $\{y_n\}$. For example, if our DSP has a single multiplier/accumulator, then the clock might trigger the following instructions:

1) multiply x_n by a_0
2) multiply x_{n-1} by a_1 and add to 1)
3) multiply x_{n-2} by a_2 and add to 2)
4) multiply y_{n-1} by b_1 and add to 3)
5) multiply y_{n-2} by b_2 and add to 4) to give y_n.

The values of x_n, x_{n-1}, x_{n-2}, y_{n-1}, y_{n-2} are stored in memory locations. You might expect that after y_n is computed, then in preparation to compute y_{n+1}, we should use a sequence of instructions to move x_{n+1} to the old x_n location, x_n to the old x_{n-1} location, x_{n-1} to the old x_{n-2} location, y_n to the old y_{n-1} location, and y_{n-2} to the old y_{n-1} location. However, especially in higher order filters, this would be a huge waste of clock cycles (instructions). Instead, a pointer is used to address the proper memory location at each clock cycle. Thus, it is not necessary to move data from memory location to memory location after computing each y_n.

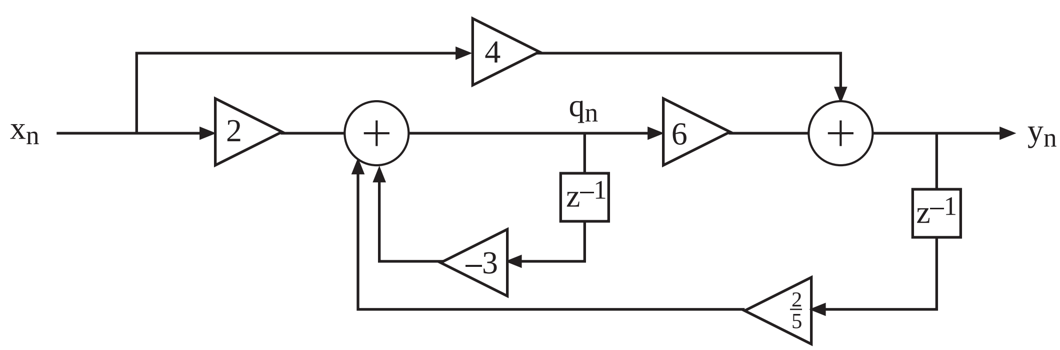
Just as there are a large number of filter structures that implement the same transfer function, there are many algorithms (for a specific DSP) that can implement a given filter structure. What are the considerations in choosing a structure/algorithm? There are generally two:

1) Speed (number of clock cycles per output)
2) Error due to finite register length.

We have not yet addressed 2). The fact that the DSP has finite-length registers, both for memory locations and the arithmetic unit, means that the digital filtering algorithm is not implemented in an exact way. There will be error at the filter output due to coefficient quantization and arithmetic roundoff. Of course, the longer the register lengths, the lower the error at the filter output. Generally, there is a tradeoff between 1) and 2). For a fixed register length, error usually can be reduced by using a more complicated (than Direct Form) filter structure, requiring more multiplications, additions, and memory locations. This in turn reduces the speed of the filter. The filter structure used in practice depends on H(z) (some transfer functions are more difficult to implement with low error), on the available register length, and on the number of clock cycles available per output.

Example

Find the transfer function of the system below and sketch a Direct Form 2 filter structure that implements the same transfer function.



We establish the intermediate quantity q_n and then write:

(i) $Y(z) = 6\ Q(z) + 4\ X(z)$

(ii) $Q(z) = 2\,X(z) - 3\,z^{-1}\,Q(z) + \frac{2}{5}\,z^{-1}\,Y(z)$

Solve (i) for Q(z) and substitute into (ii):

$$\text{(i)} \;\Rightarrow\; Q(z) = \frac{1}{6}\,Y(z) - \frac{2}{3}\,X(z)$$

$$\text{(ii)} \;\Rightarrow\; \frac{1}{6}\,Y(z) - \frac{2}{3}\,X(z) = 2\,X(z) - 3\,z^{-1}\left[\frac{1}{6}Y(z) - \frac{2}{3}\,X(z)\right] + \frac{2}{5}\,z^{-1}\,Y(z)$$

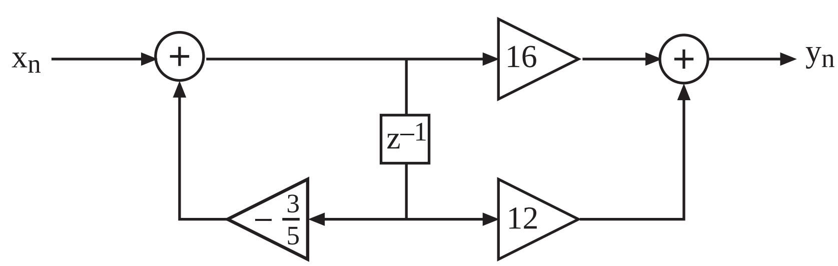
$$\Rightarrow\; Y(z)\left[\frac{1}{6} + \frac{1}{2}\,z^{-1} - \frac{2}{5}\,z^{-1}\right] = X(z)\left[\frac{2}{3} + 2 + 2\,z^{-1}\right]$$

$$\Rightarrow\; H(z) = \frac{\frac{8}{3} + 2\,z^{-1}}{\frac{1}{6} + \frac{1}{10}\,z^{-1}}$$

Now, the quickest way to map H(z) into a Direct Form filter structure is to first normalize the denominator of H(z) to have a leading term equal to one. Thus, multiply both the numerator and denominator of H(z) by 6 to give

$$H(z) = \frac{16 + 12\,z^{-1}}{1 + \frac{3}{5}\,z^{-1}}$$

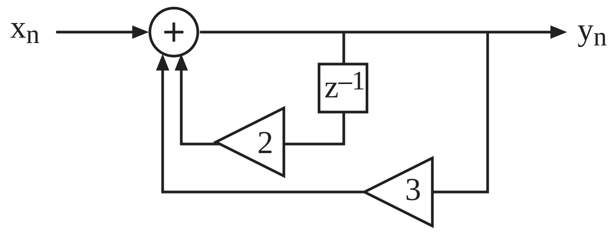
The Direct Form 2 structure having this transfer function is then



This structure is far simpler than the previous one and it computes exactly the same output $\{y_n\}$.

<u>Important Note</u>: Digital filter structures cannot have delay-free loops.

Example Consider the filter structure

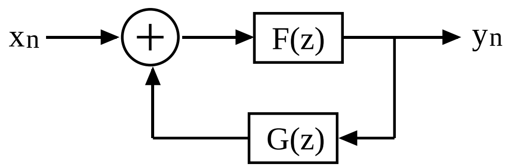


This system is unrealizable because

$$y_n = x_n + 2\, y_{n-1} + 3\, y_n$$

Since the system is clocked and the elements of $\{y_n\}$ are computed one at a time, we cannot have element y_n depend on itself as in the above equation.

A handy fact:



$$\Rightarrow H(z) = \frac{F(z)}{1 - F(z)G(z)}$$

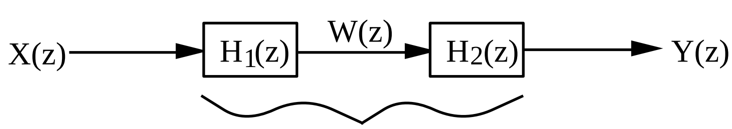
Proof:

$$Y(z) = F(z)\,[X(z) + G(z)\,Y(z)]$$

$$\Rightarrow\; Y(z)\,[1 - F(z)\,G(z)] = F(z)\,X(z)$$

$$\Rightarrow\; H(z) = \frac{Y(z)}{X(z)} = \frac{F(z)}{1 - F(z)G(z)} \qquad \checkmark$$

Cascade (series) and **parallel** connections:



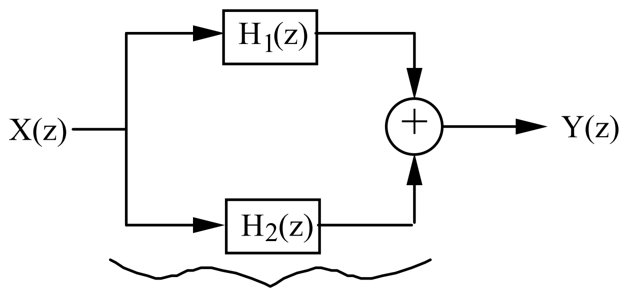
TF = $H_1(z)\,H_2(z)$

Proof:

$$Y(z) = H_2(z)\,W(z)$$

$$= H_2(z)\,[H_1(z)\,X(z)]$$

$$\Rightarrow H(z) = \frac{Y(z)}{X(z)} = H_1(z)\, H_2(z) \quad ✓$$



Proof:

$$Y(z) = H_1(z)\, X(z) + H_2(z)\, X(z)$$

$$= \left[H_1(z) + H_2(z)\right] X(z)$$

$$\Rightarrow H(z) = \frac{Y(z)}{X(z)} = H_1(z) + H_2(z) \quad ✓$$

Complex Systems

In a previous lecture, we pointed out that systems with complex-valued inputs, outputs, adders, and multipliers are realizable. That is, they are implemented using real adders, real multipliers, and real delays. The following example gives further insight into how this can be done.

Example

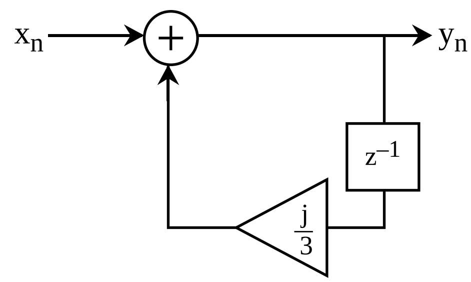
Draw a block diagram of a system that implements $y_n = h_n * x_n$ where $\{x_n\}$ and $\{y_n\}$ are complex-valued and $h_n = \left(\frac{j}{3}\right)^n u_n$. All adders, multipliers, and delays should be real.

<u>Solution</u>

We have

$$H(z) = \frac{z}{z - \frac{j}{3}} = \frac{1}{1 - \frac{j}{3} z^{-1}}$$

So, we might consider



Here, though, $\{x_n\}$ and $\{y_n\}$ are each pairs of real-valued sequences. Write $x_n = (x_R(n), x_I(n))$ and $y_n = (y_R(n), y_I(n))$. Then, recalling the definitions of complex multiplication and addition, we have

$$\frac{j}{3} y_{n-1} = \left(0, \frac{1}{3}\right) \bullet (y_R(n-1), y_I(n-1)) = \left(-\frac{1}{3} y_I(n-1), \frac{1}{3} y_R(n-1)\right) .$$

Then

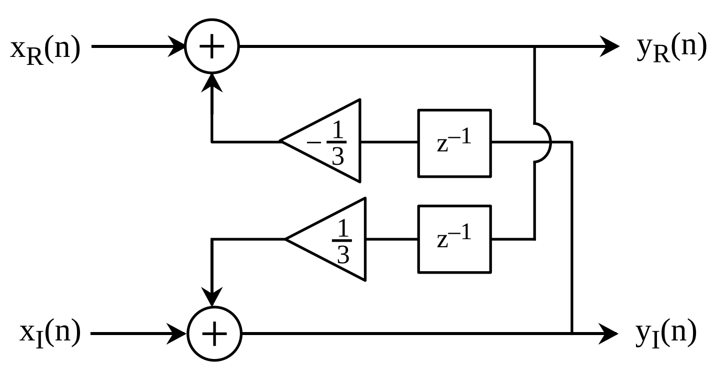
$$\begin{aligned} y_n &= (y_R(n), y_I(n)) = x_n + \frac{j}{3} y_{n-1} \\ &= (x_R(n), x_I(n)) + \left(-\frac{1}{3} y_I(n-1), \frac{1}{3} y_R(n-1)\right) \\ &= \left(x_R(n) - \frac{1}{3} y_I(n-1), x_I(n) + \frac{1}{3} y_R(n-1)\right) \end{aligned}$$

Thus,

$$y_R(n) = x_R(n) - \frac{1}{3} y_I(n-1)$$

$$y_I(n) = x_I(n) + \frac{1}{3} y_R(n-1)$$

These last two equations tell us exactly how to implement the system:



This is a physical implementation of the previous block diagram.

Approaching the implementation problem in an alternate way, we can find a more complicated, but equivalent, physical realization. Write

$$
\begin{aligned}
y_R(n) + j\,y_I(n) &= \big(h_R(n) + j\,h_I(n)\big) * \big(x_R(n) + j\,x_I(n)\big) \\
&= \big(h_R(n) * x_R(n) - h_I(n) * x_I(n)\big) + j\big(h_R(n) * x_I(n) + h_I(n) * x_R(n)\big) \qquad (\Delta)
\end{aligned}
$$

Furthermore, we can write

$$H(z) = H_R(z) + j\,H_I(z)$$

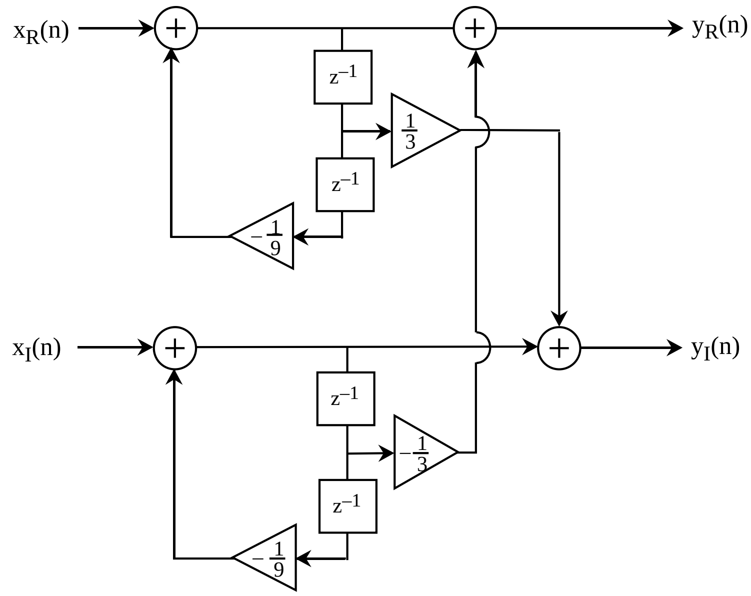
where $H_R(z)$ is the z-transform of $h_R(n)$ and $H_I(z)$ is the z-transform of $h_I(n)$. Since both $h_R(n)$ and $h_I(n)$ are real-valued, the coefficients of both $H_R(z)$ and $H_I(z)$ must be real-valued. How do we find $H_R(z)$ and $H_I(z)$? There are two ways. The easiest is to write

$$
\begin{aligned}
H(z) &= \frac{z}{z - \frac{j}{3}} = \frac{z}{z - \frac{j}{3}}\,\frac{z + \frac{j}{3}}{z + \frac{j}{3}} = \frac{z^2 + j\frac{1}{3}z}{z^2 + \frac{1}{9}} \\
&= \frac{1}{1 + \frac{1}{9}z^{-2}} + j\,\frac{\frac{1}{3}z^{-1}}{1 + \frac{1}{9}z^{-2}}
\end{aligned}
$$

Thus,

$$H_R(z) = \frac{1}{1 + \frac{1}{9}z^{-2}}\,, \qquad H_I(z) = \frac{\frac{1}{3}z^{-1}}{1 + \frac{1}{9}z^{-2}} \qquad (\Delta\Delta)$$

Using this, with Eq. (Δ) above, our implementation of $H(z)$ has two copies of $H_R(z)$ and two copies of $H_I(z)$, with inputs $x_R(n)$ and $x_I(n)$. The outputs of the copies of $H_R(z)$ and $H_I(z)$ are then interconnected to produce $y_R(n)$ and $y_I(n)$. Since $H_R(z)$ and $H_I(z)$ are nearly the same in this example, however, the diagram can be simplified to



Although this diagram is quite different from our earlier implementation, it is equivalent in the sense that it computes the same $y_R(n)$ and $y_I(n)$.

Note: (ΔΔ) can be derived in an alternate, but lengthier way. Since $h_n = \left(\frac{j}{3}\right)^n u_n$, we have

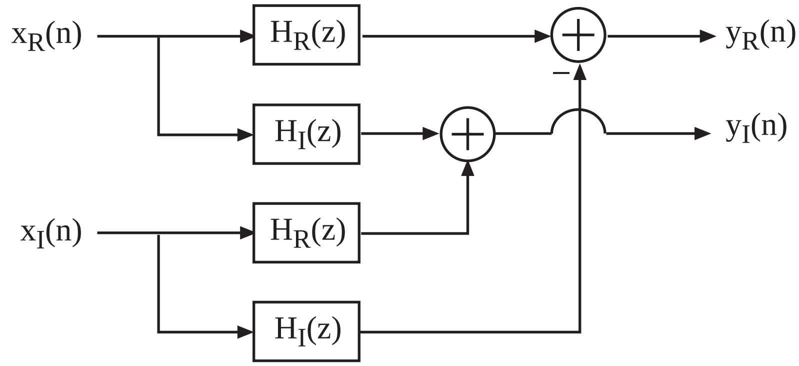
$$h_R(n) = 1 \quad 0 \quad \frac{-1}{9} \quad 0 \quad \frac{1}{81} \quad 0 \quad \ldots$$
$$h_I(n) = 0 \quad \frac{1}{3} \quad 0 \quad \frac{-1}{27} \quad 0 \quad \frac{1}{243} \quad \ldots$$

Taking z-transforms of these sequences gives (ΔΔ).

In general

$$\begin{aligned} Y(z) &= \left(H_R(z) + jH_I(z)\right)\left(X_R(z) + jX_I(z)\right) \\ &= \left(H_R(z)X_R(z) - H_I(z)X_I(z)\right) + j\left(H_R(z)X_I(z) + H_I(z)X_R(z)\right) \end{aligned}$$

so that a possible implementation is always



System Analysis

As we have seen, the input-output relationship of a linear-shift invariant (LSI) system is captured through its response to a single input, that due to a discrete-time impulse, or the impulse response of the system. There are a number of important properties of LSI systems that we can study by observing properties of its impulse response directly. Perhaps one of the more important properties of such systems is whether or not they are stable, that is, whether or not the output of the system will remain bounded for all time when the input to the system is bounded for all time. While for continuous-time systems and circuits stability may be required for ensuring that components do not become damaged as voltages or currents grow unbounded in a system, for discrete-time systems, stability can be equally important. For example, practical implementations of many discrete-time systems involve digital representations of the signals. To ensure proper implementation of the operations involved, the numerical values of the signal levels must remain within the limits of the number system used to represent the signals. If the signals are represented using fixed-point arithmetic, there may be strict bounds on the dynamic range of the signals involved. For example, any real number $-1 \le x \le 1$ can be represented as an infinite binary string in two's complement notation as

$$x = -b_0 + \sum_{k=1}^{\infty} b_k 2^{-k} .$$

In a practical implementation, only finite-precision representations are available, such that all values might be represented and computed using fixed-point two's compliment arithmetic where any signal at a given point in time would be represented as a B+1-bit binary string $-1 \le x[n] < 1$,

$$x[n] = -b_0 + \sum_{k=1}^{B} b_k 2^{-k} .$$

Now, if the input signal such a system was carefully conditioned such that it was less than 1 in magnitude, it is important that not only does the output remain less than 1 in magnitude, but also all intermediate calculations must also. If not, then the numbers would overflow, and produce incorrect results, i.e. they would not represent the true output of the LSI system to the given input. If the discrete-time system were used to control a mechanical system such as an aircraft, such miscalculations due to instability of the discrete-time system could produce erratic or even catastrophic results.

Stability

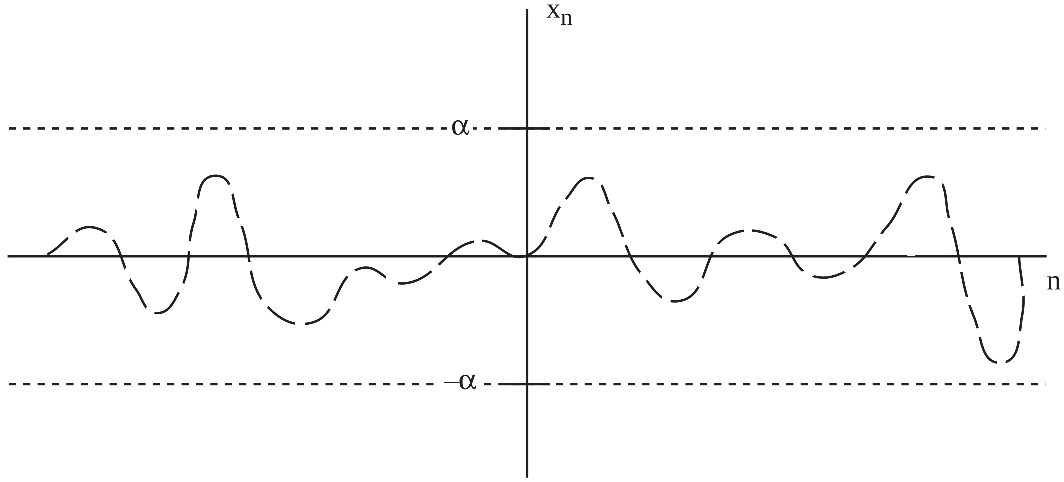
A system is bounded-input, bounded-output (BIBO) stable if for every bounded input, the resulting output is bounded. That is, if there exists a fixed positive constant α, such that

$$|x[n]| \le \alpha < \infty, \text{ for all } n,$$

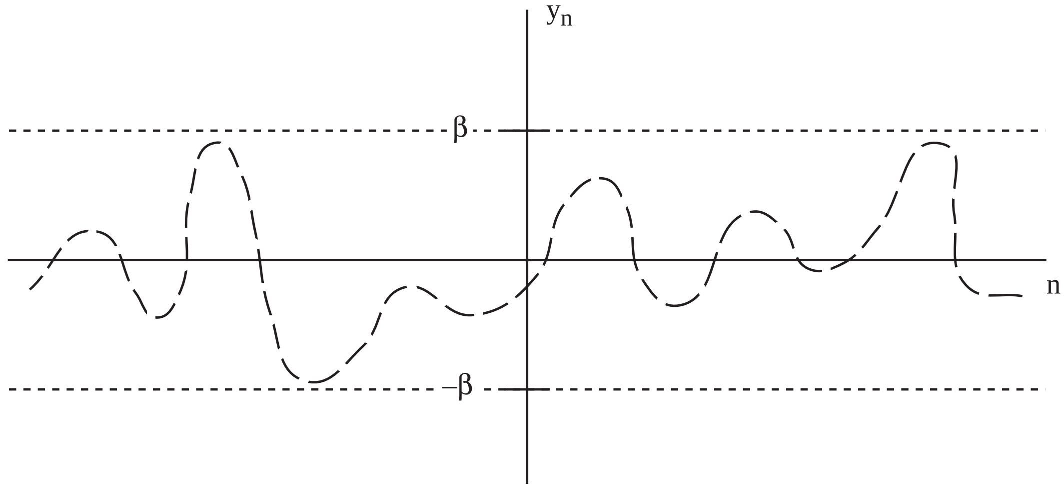
then there exists a fixed positive constant β, such that

$$|y[n]| \le \beta < \infty, \text{ for all } n,$$

where the constants α and β fixed, i.e. they do not depend on n. Pictorially, if every bounded $x[n]$:



causes a bounded $y[n]$:



then system is BIBO stable. Note that BIBO stability is a property of the system and not the inputs or outputs. While it may be possible to find specific bounded inputs such that the outputs remain bounded, a system is only BIBO stable if the output remains stable for all possible inputs. If there exists even one input for which the output grows unbounded, then the system is not stable in the BIBO sense.

How do we check if a system is BIBO stable? We cannot possibly try every bounded input and check that the resulting outputs are bounded. Rather, the input-output relationship must be used to prove that BIBO stability holds. Similarly, the following theorems can be used to provide simple tests for BIBO stability.

Theorem 1 An LSI system with impulse response $h[n]$ is BIBO stable if and only if the impulse response is absolutely summable. That is, the system is BIBO stable if and only if

$$\sum_{n=-\infty}^{\infty} \left|h[n]\right| < \infty \ .$$

Proof: First, to prove sufficiency, we must show that absolute summability implies BIBO stability. That is, given that $\sum_{n=-\infty}^{\infty}\left|h[n]\right| < \gamma < \infty$, we for any $x[n]$ and real-valued, positive $\alpha < \infty$ satisfying $\left|x[n]\right| < \alpha$ for all n, we have from the convolution sum

$$\begin{aligned} y[n] &= \left| \sum_{m=-\infty}^{\infty} h[m]x[n-m] \right| \\ &\le \sum_{m=-\infty}^{\infty} \left| h[m]x[n-m] \right| \\ &= \sum_{m=-\infty}^{\infty} \left| h[m] \right| \left| x[n-m] \right| \\ &\le \alpha \sum_{m=-\infty}^{\infty} \left| h[m] \right| \\ &\le \alpha\gamma < \infty \end{aligned}$$

which implies that the output is bounded, i.e., $\left| y[n] \right| < \beta = \alpha\gamma < \infty$ for all n. To prove necessity, we need to show that when the impulse response is not absolutely summable, then there exists a sequence $x[n]$ that is bounded, but for which the output of the system is not bounded. That is, given $\sum_{m=-\infty}^{\infty} \left| h_m \right| = \infty$, we need to show that there exists a bounded sequence $x[n]$ that produces an output $y[n]$ such that $y[n_0] = \infty$ for some fixed n_0, i.e., $y[n]$ is not bounded. From the convolution sum, we have

$$y[n_0] = \sum_{m=-\infty}^{\infty} x[m]h[n_o - m].$$

By selecting the sequence $x[n]$ to be such that $x[m] = h^*[n_0 - m] / | h[n_0 - m] |$, (for real-valued $h[n]$, this amounts to $x[m] = \mathrm{sgn}(h[n_0 - m]) = \pm 1$), then we have that

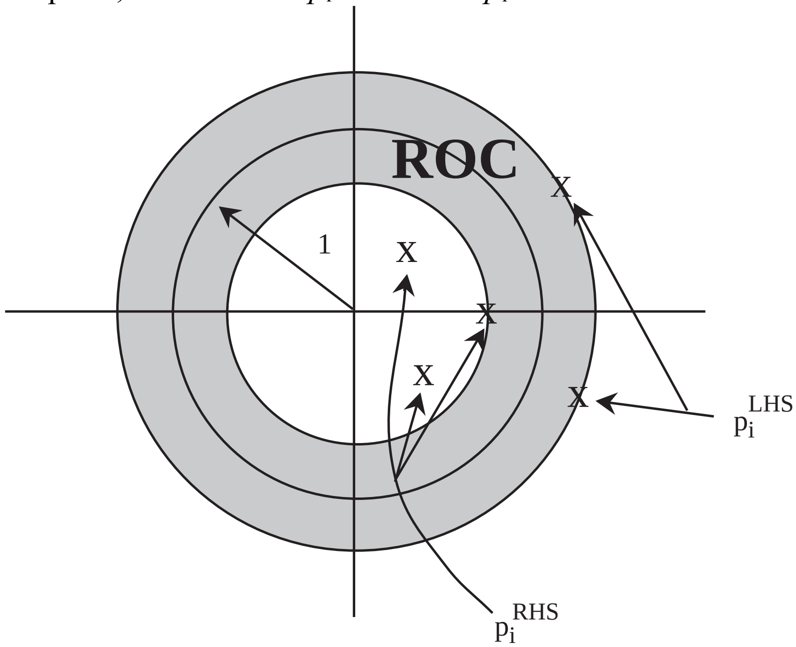
$$\begin{aligned} y[n_0] &= \sum_{m=-\infty}^{\infty} x[m]h[n_0 - m] \\ &= \sum_{m=-\infty}^{\infty} \frac{h^*[n_0 - m]h[n_0 - m]}{| h[n_0 - m] |} \\ &= \sum_{m=-\infty}^{\infty} | h[n_0 - m] |, \quad \text{letting } k = n_0 - m \\ &= \sum_{k=-\infty}^{\infty} | h[k] | = \infty \end{aligned}$$

which shows that the output is unbounded, completing the proof.

BIBO stablility of a system can also be directly determined from the transfer function $H(z)$, relating the z-transform of the input to the z-transform of the output.

Theorem 2 An LSI system with a rational transfer function (in minimal form) is BIBO stable if and only if its region of convergence, ROC_H, includes the unit circle.

Proof: First, to prove sufficiency, assume the region of convergence ROC_H includes the unit circle. Next, to illustrate that this implies absolute summability, i.e. $\sum_{n=-\infty}^{\infty} | h[n] | < \infty$, we consider the poles of the system function in two groups. First, the poles (roots of the denominator polynomial) that lie inside the unit circle, i.e. those that correspond to right-sided inverse z-transforms, are labeled p_i^{RHS} in the figure below. Second, the poles that lie outside the unit circle correspond to left-sided inverse z-transforms and are labled p_i^{LHS} in the figure below. For these two sets of poles, we have that $p_i^{RHS} < 1$ and $p_i^{LHS} > 1$.



The inverse z-transform, as determined using partial fraction expansion has the form

$$h_n = \quad h[n] = \begin{cases} \sum_{i=1}^{K} a_i \left(p_i^{RHS}\right)^n & n \geq 0 \\ \sum_{i=1}^{L} b_i \left(p_i^{LHS}\right)^n & n < 0 \end{cases}$$

$$= \sum_{i=1}^{K} a_i \left(p_i^{RHS}\right)^n u[n] + \sum_{i=1}^{L} b_i \left(p_i^{LHS}\right)^n u[-n-1]$$

when there are no repeated poles (the results extend easily to the repeated-roots case). Since we have that $|p_i^{LHS}| > 1$ and $|p_i^{RHS}| < 1$, we have that

$$\begin{aligned}
\sum_{n=-\infty}^{\infty} |h[n]| &= \sum_{n=-\infty}^{\infty} \sum_{i=1}^{K} a_i \left(p_i^{RHS}\right)^n u[n] + \sum_{i=1}^{L} b_i \left(p_i^{LHS}\right)^n u[-n-1] \\
&= \sum_{n=0}^{\infty} \left| \sum_{i=1}^{K} a_i \left(p_i^{RHS}\right)^n \right| + \sum_{n=-\infty}^{-1} \left| \sum_{i=1}^{L} b_i \left(p_i^{LHS}\right)^n \right| \\
&\leq \sum_{n=0}^{\infty} \sum_{i=1}^{K} |a_i| \left|p_i^{RHS}\right|^n + \sum_{n=-\infty}^{-1} \sum_{i=1}^{L} |b_i| \left|p_i^{LHS}\right|^n \\
&= \sum_{i=1}^{K} \frac{|a_i|}{1 - p_i^{RHS}} + \sum_{i=1}^{L} \left(\frac{|b_i|}{1 - p_i^{LHS}} - 1 \right) < \infty
\end{aligned}$$

To show necessity, assume BIBO stability, and hence absolute summability of the impulse response, and then, for any point z on the unit circle, we have that

$$\left|H(z)\right|\Big|_{|z|=1} = \left|\sum_{n=-\infty}^{\infty} h[n]\,z^{-n}\right|\Bigg|_{|z|=1}$$
$$\leq \sum_{n=-\infty}^{\infty} \left|h[n]\right|\left|z^{-n}\right|\Big|_{|z|=1}$$
$$= \sum_{n=-\infty}^{\infty} \left|h[n]\right|\,|\,1\,|^{-n}$$
$$= \sum_{n=-\infty}^{\infty} \left|h[n]\right| < \infty,$$

which implies that the region of convergence includes the unit circle and completes the proof.

Corollary of Theorem 2: A causal LSI system with a rational transfer function (in minimal form) is BIBO stable if and only if all of its poles are inside the unit circle.

Proof: A causal system has all poles between origin and ROC_H with at least one pole on the inner radius of ROC_H. So, all of the poles are inside unit circle if and only if ROC_H includes the unit circle. However, by Theorem 2, ROC_H includes the unit circle if and only if the system is BIBO stable.

These properties are explored in the following examples.

Example
Consider the following LSI system with impulse response $h[n]$, we have that

$$h[n] = (\cos\theta\, n)u[n]$$
$$\sum_{n=-\infty}^{\infty} |\,h[n]\,| = \sum_{n=0}^{\infty} |\cos\theta\, n\,|$$
$$= \infty.$$

Therefore, the system is not BIBO stable.

Example
Consider the following transfer function for a causal LSI system,

$$H(z) = \frac{z^2 - 3z + 2}{z^3 - 2z^2 + \frac{z}{2} - 1}.$$

Factoring the denominator, we have that

$$H(z) = \frac{(z-1)(z-2)}{\left(z^2 + \frac{1}{2}\right)(z-2)} = \frac{z-1}{z^2 + \frac{1}{2}},$$

which has poles at $\pm j/\sqrt{2}$. The system is therefore causal and has all of its poles inside the unit circle. Therefore the system is BIBO stable. Note that as done in this example, factors that are common to the numerator and denominator must be cancelled before applying the stability test.

Example
Consider the following system function of an LSI system,

$$H(z)=\frac{z}{z+100}, |z|<100 .$$

Since ROC_H includes the unit circle, the system therefore must be BIBO stable. In this example, the impulse response $h[n]$ happens to be left-sided.

Example

Consider the following impulse response of an LSI system,

$$h[n]=\begin{cases} 4^n & 0 \le n \le 10^6 \\ n\left(\frac{1}{2}\right)^n & 10^6+1 \le n < \infty \\ 0 & n<0 \end{cases}$$

Testing for absolute summability, we have that

$$\sum_{n=-\infty}^{\infty} |h_n| = \sum_{n=0}^{10^6} 4^n + \sum_{n=10^6+1}^{\infty} n\left(\frac{1}{2}\right)^n \quad < \quad \infty,$$

therefore the system is indeed BIBO stable.

We continue exploring the properties of LSI systems through observation of their system functions (that is, the z-transform of the impulse response), with a focus on the relationship between the region of convergence of the z-transform and the stability and causality of the system.

Example

Consider the following system function of a stable LSI system,

$$H(z)=\frac{z}{\left(z-\frac{1}{4}\right)(z-2)} .$$

Can system be causal?

Answer: No, it cannot be causal. First, note that although the region of convergence is not explicitly stated, it is implicitly determined. Noting that the system is stable, we know that the region of convergence must include the unit circle. Given the pole locations, we know that the region of convergence must be $\{z : \frac{1}{4} < |z| < 2\}$ implying that the impulse response will have left-sided and right-sided components and that $h[n]$ must be two-sided. Since the impulse response is two-sided, this implies that the system cannot be causal, i.e. $h[n]$ is non-zero for $n < 0$ and from the convolution sum,

$$y[n]=\sum_{m=-\infty}^{\infty} h[m]x[n-m],$$

we see that $y[n]$ will depend on future values of the input $x[n]$.

Example

Consider the following discrete time system,

$$y[n] = (x[n])^2 .$$

Is it stable?

Answer: This system is not linear. Therefore, we cannot apply a stability test involving either the unit-pulse response or transfer function, since the tests discussed so far apply only to LSI systems. Since this system is not LSI, the convolution sum does not hold, so that the input output relationship does not satisfy $y[n]=x[n]*h[n]$ or $Y(z) = H(z)\,X(z)$. Instead, we appeal to the definition of BIBO stability. Suppose that $x[n]$ is bounded, then we have

$$|x[n]| < \alpha < \infty$$

and subsequently

$$|y[n]| = |x[n]|^2 < \alpha^2 < \infty .$$

Therefore, we have shown that any bounded input produces a bounded output and that the system is BIBO stable.

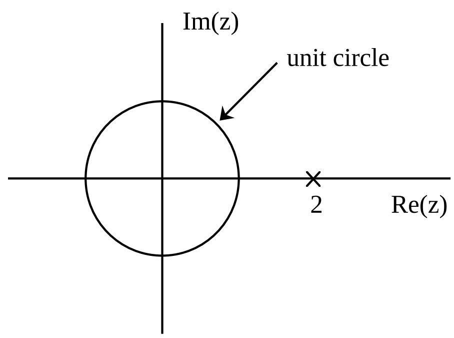
Unbounded Outputs

Given an unstable LSI system, how do we find a bounded input that will cause an unbounded output? This will be illustrated by example for some causal systems in the following examples.

Example

Consider the following causal LSI system with pole-zero plot shown to the right and with system function $H(z)$ given by

$$H(z) = \frac{z}{z-2}, |z| > 2$$

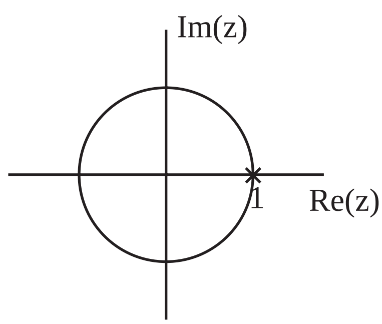


The impulse response is therefore given by $h[n] = 2^n\, u[n]$ and is itself unbounded. Since $h[n]$ grows without bound, almost any bounded input will cause the output to be unbounded. For example, taking $x[n] = \delta[n]$ would yield $y[n] = h[n]$.

Example

Now consider the following LSI system with pole-zero plot and system function given right and below, respectively.

$$H(z) = \frac{z}{z-1}, |z| > 1$$



Although the system is not stable, the impulse response remains bounded, as $h[n] = u[n]$, in this case. Here we could choose $x[n] = u[n]$ (which is bounded) so that $y[n]$ will be a linear ramp in time. Looking at the z-transform of the output, this corresponds to forcing $Y(z)$ to have a double pole at $z = 1$, i.e.

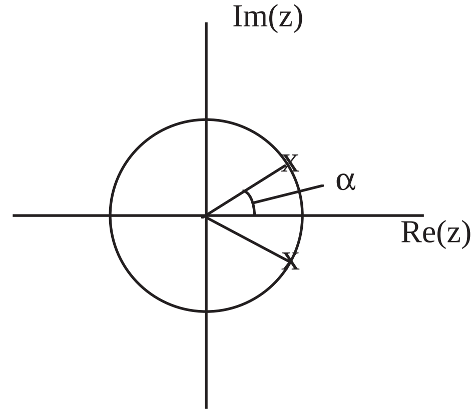
$$Y(z) = H(z)X(z) = \frac{z^2}{(z-1)^2},$$

which for the region of convergence of this output corresponds to a sequence that grows linearly in time.

Example

Here we consider an LSI system with a complex-conjugate pole pair on the unit circle.

$$H(z) = \frac{z^2 - z\cos\alpha}{(z-e^{j\alpha})(z-e^{-j\alpha})}, |z| > 1$$



The complex conjugate pair of poles on the unit circle corresponds to a sinusoidal oscillating impulse response

$$h[n] = \cos(\alpha n)\, u[n].$$

Thinking of the z-transform of the output, note that choosing $x[n] = h[n]$ will cause $Y(z)$ to have double poles at $z = e^{\pm j\alpha}$, which will in turn cause $y[n]$ to have the form of n times $\cos \alpha n$, which is unbounded.

From these examples with causal systems, we see that for systems with poles outside the unit circle, since the impulse response itself grows unbounded, substantial effort would be required to find a bounded input that will not cause an unbounded output. For poles on the unit circle, it is more difficult to find bounded inputs that ultimately cause the output to be unbounded. In some fields, such as dynamic systems or control, LSI systems with poles on the unit circle are called "marginally stable" systems. In our terminology, they are simply unstable systems.

Impulse Distributions

To continue our discussion of discrete-time systems and aid in the development of the discrete-time Fourier transform, it will be convenient to recall some of the properties of continuous-time impulse distributions. While much of this discussion will be review, it is important to recall the "behavioral definition" of continuous-time impulses. That is, an impulse is defined only by what it does inside of an integral and cannot be considered to be a function in and of itself.

Definition. A *distribution* is a mapping from a function to a number.

Definition The *impulse* δ is the distribution:

$$\delta\left[f(t)\right] \triangleq f(0), \qquad (1)$$

where $f(t)$ may be any continuous function of the real variable t. We often write the distribution δ as $\delta(t)$ and use the notation

$$\int_{-\infty}^{\infty} \delta(t)\, f(t)\, dt \triangleq f(0). \qquad (2)$$

As a special case, when the function $f(t) = 1$, we have that

$$\int_{-\infty}^{\infty} \delta(t) dt = 1.$$

The use of impulse distributions within expressions that contain functions and integrals, however, might give the mistaken impression that they are in fact functions and that these integrals are well-defined. However it is important to remember that such expressions are in fact not integrals at all, and that the integral calculus cannot be applied or assumed to hold in expressions containing impulses. It is important to recall that

a) $\delta(t)$ is not a function.
b) The integral sign in (2) is not an integral in any meaningful sense.
c) Equation (2) is just alternate notation for the more explicit notation in Equation (1).

Once the notion of a distribution is understood, it can be generalized to a broader family of impulse distributions, such as

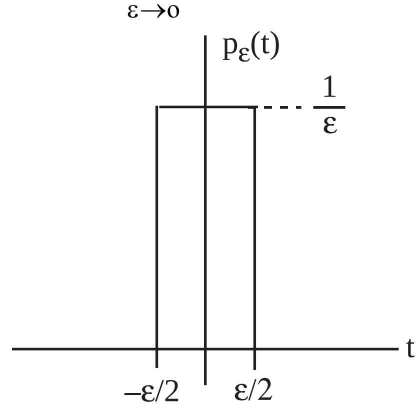
$$\delta_{t_0}\left[f(t)\right] \triangleq f(t_0). \qquad (3)$$

The notation usually used involves placing the impulse distribution under an integral, as in

$$\int_{-\infty}^{\infty} \delta(t-t_0) f(t) dt \triangleq f(t_0) \qquad (4)$$

however, once again, equation (4) is just alternate notation for that of equation (3). Engineering intuition might be obtained in some contexts by *thinking* of $\delta(t)$ as the limit of a sequence of tall, narrow pulses, each having area = 1, e.g.,

$$\delta(t) = \lim_{\varepsilon \to o} p_\varepsilon(t) \text{ with}$$



However, in fact, this limit does not exist at $t = 0$; and therefore δ is *not* a function. However, we might obtain some intuition by considering what happens (for well-behaved functions $f(t)$) to the following integral

$$f(0) = \lim_{\varepsilon \to o} \int_{-\infty}^{\infty} p_\varepsilon(t) f(t) dt = \int_{-\infty}^{\infty} \delta(t) f(t) dt\,, \tag{5}$$

where the second integral in equation (5) is interpreted as defined in equation (2). Similarly, we can write

$$f(t_0) = \lim_{\varepsilon \to 0} \int_{-\infty}^{\infty} p_\varepsilon(t - t_0) f(t) dt = \int_{-\infty}^{\infty} \delta(t - t_0) f(t) dt\,, \tag{6}$$

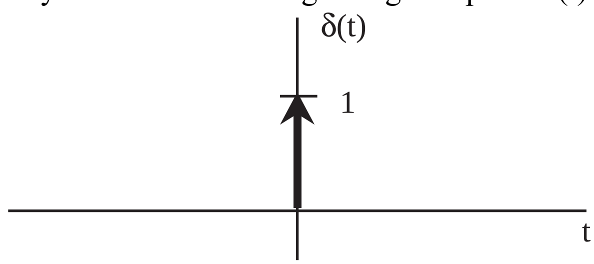
where the second integral in equation (6) is again as defined in equation (4). By attempting to *define* the impulse distribution as a function determined by the limit of the sequence of functions

$$\delta(t) = \lim_{\varepsilon \to 0} p_\varepsilon(t), \tag{7}$$

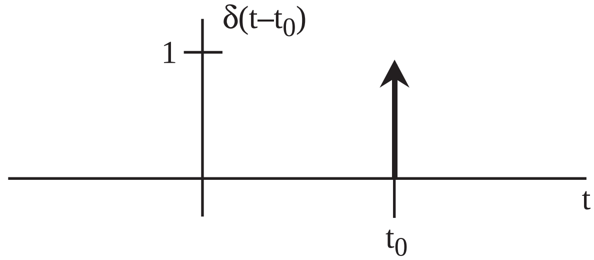
or

$$\delta(t - t_0) = \lim_{\varepsilon \to o} p_\varepsilon(t - t_0), \tag{8}$$

and then applying that definition to the integrals in equation (2) or (4), in comparison with equations (5) and (6), would be to bring the limit inside the integral. This is mathematically incorrect and is an abuse of notation. Equations (7) and (8) actually imply (i.e. should be interpreted as) equations (5) and (6), respectively. To facilitate working with impulse distributions, although they are not functions, it is convenient to have a symbolic notation that enables visualizing distributions in a manner similar to how we plot functions. Since they are not functions, and cannot be interpreted as such, they cannot be plotted. However, the following graphical notation is typically used for visualizing a single impulse $\delta(t)$:



where the number written above the vertical arrow indicates the scale factor that is applied to $f(0)$. This can also be thought of as the "area" of the impulse, when written in integral notation. The vertical arrow is placed at the location from which the distribution selects the value of the function to which it is applied. As such, $\delta(t - t_o)$ is pictured as



here, the "area" of the impulse is indicated by the height of the impulse located at the time t_o.

Example

As an example, we could consider the following distribution,

$$g(t) = \frac{1}{2}\ \delta(t) + \delta(t-1) + \frac{3}{2}\ \delta(t+1),$$

where the distribution $g(t)$ needs to be again treated as a mapping from a function to a number. We have

$$\begin{aligned}\int_{-\infty}^{\infty} g(t)\, f(t)\, dt &= \int_{-\infty}^{\infty} (\tfrac{1}{2}\delta(t) + \delta(t-1) + \tfrac{3}{2}\delta(t+1))\, f(t)\, dt \\ &= \int_{-\infty}^{\infty} \tfrac{1}{2}\delta(t) f(t)\, dt + \int_{-\infty}^{\infty} \delta(t-1) f(t)\, dt + \int_{-\infty}^{\infty} \tfrac{3}{2}\delta(t+1) f(t)\, dt \\ &= \tfrac{1}{2} f(0) + f(1) + \tfrac{3}{2} f(-1).\end{aligned}$$

Graphically, $g(t)$ can be pictured as follows: