IIR Filters: An Overview

Topics: IIR filters, Z-transform, poles and zeros.

A digital filter's impulse response is the output when the input is the impulse function $\delta[n]$, as indicated in Figure 1.

Figure 1: Input-output relationship for a digital produces the impulse response when the input is an impulse function.

We've already seen one kind of impulse response, that of a FIR filter. If the filter is specified by

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

then the impulse response is simply the filter coefficients:

$$h[n]=\begin{cases} b_n, & n=0,\ldots,M-1 \\ 0, & else \end{cases}$$

This is of course why the filter is called a finite (duration) impulse response filter.

Another type of filter is specified by the input-output equation

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

In this expression, not only are inputs x[n] being used, but so too are past outputs y[n]. The first sum, using coefficients b_k, is called the "feedforward" portion, and the second sum, using coefficients a_k, is called the "feedback" portion. The use of past outputs, or feedback, adds significantly to the possibilities of a digital filter.

Because this digital filter uses past outputs, we need to assume initial conditions. We use the "initial rest" assumption, which requires that

$$x[n]=y[n]=0, \ n<0.$$

Example:

Consider the filter

$$y[n]=a_1 y[n-1]+x[n]$$

The impulse response of this filter is the output when $x[n]=\delta[n]$. We find that

$$h[0]=1, \ h[1]=a_1, \ h[2]=a_1^2, \ h[3]=a_1^3, \ldots h[n]=a_1^n$$

This impulse is infinite in duration. Any filter whose impluse response is of infinite-duration is called an IIR filter. An example is shown in Figure 2.

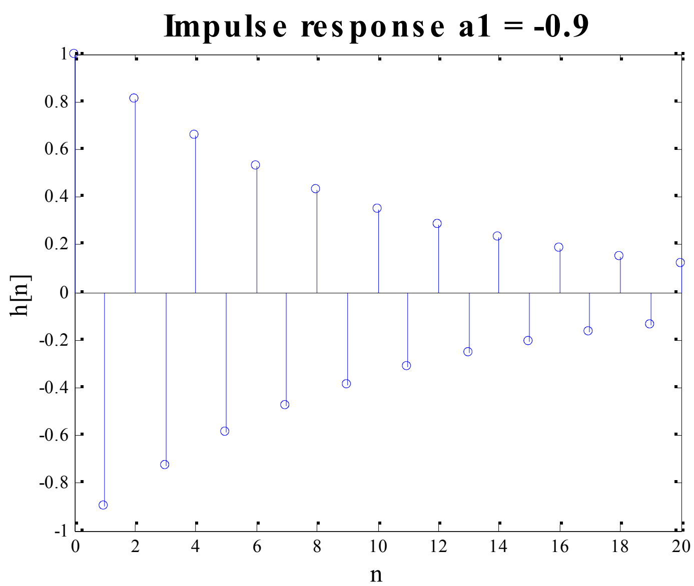

Figure 2: The impulse response of the filter in the example with $a_1=-0.9$.

IIR filters are not much more difficult to implement in software than FIR filters. For example, consider the filter

$$y[n]=a_1 y[n-1]+a_2 y[n-2]+x[n]+b_1 x[n-1]$$

The psuedocode for implementing this filter is shown in Figure 3:

```
yprev1 = 0;        yprev2 = 0;
xprev1 = 0;

/* get x from A/D convertor and compute y */
repeat {

        y = a1*yprev + a2*yprev2 + x + b1*xprev;

        /* update filter state */
        xprev = x;
        yprev2 = yprev1;  yprev1 = y;

} until (no more inputs)
```

Figure 3: Psuedocode for implementing IIR filter described above.

Stability

Because IIR filters use feedback, they are not necessarily stable. A *stable* filter is one for which any input that is limited in amplitude is guaranteed to produce an output that is also limited in amplitude, though the limits on the input and output can different numbers.

An example of an *unstable* filter is as follows:

$$y[n]=-1.2\,y[n-1]+x[n]$$

The factor of -1.2 multiplying the feedback term produces an unbounded impulse response. We see that (see Figure 4)

$$h[0]=1,\ h[1]=-1.2,\ h[2]=1.44,\ h[3]=-1.728,\ldots,\ h[10]=6.19,\ldots$$

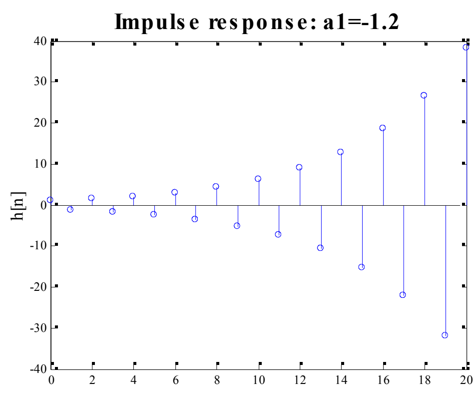

Figure 4: Impulse response of an unstable filter.

If we want to design an IIR filter, how can we be sure that it is stable? The theory of the Z-transform is designed for this purpose.

Z-transform

The z-transform is motivated by the fact that since we know a lot about polynomials from algebra, it would be useful to transform IIR filters into polynomials.

Suppose $z = x + jy$ is a complex number. The z-transform is defined as the power series

$$\boldsymbol{X}(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

A very important property of the z-transform is that a delay in the signal is transformed into a multiplication by z^{-1} :

$$y[n] = x[n-1] \rightarrow \boldsymbol{Y}(z) = \sum_{n=-\infty}^{\infty} x[n-1] z^{-n} = \sum_{k=-\infty}^{\infty} x[k] z^{-(k+1)} = \boldsymbol{X}(z) z^{-1}$$

This property allows us to transform both sides of the IIR filter equation

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

We obtain

$$\boldsymbol{Y}(z) = \sum_{k=0}^{M-1} b_k \boldsymbol{X}(z) z^{-k} + \sum_{k=1}^{N} a_k \boldsymbol{Y}(z) z^{-k}$$

Collecting common terms, we see that

$$\boldsymbol{Y}(z) = \boldsymbol{H}(z) \boldsymbol{X}(z)$$

where

$$\boldsymbol{H}(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_{M-1} z^{-(M-1)}}{1 - a_1 z^{-1} - \cdots - a_N z^{-N}}$$

This function $\boldsymbol{H}(z)$ encapsulates the properties of the filter in a ratio of polynomials. It is known as the *transfer function* of the filter.

The z-transform of $x[n]=\delta[n]$ is $\boldsymbol{X}(z)=1$. When the input is the impulse function, the output $y[n]$ is the impulse response $h[n]$. Therefore, the transfer function has another interpretation, as the z-transform of the impulse response:

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

The numerator and denominator of the transfer function of an IIR filter are both polynomials, and all polynomials have roots. The roots of the numerator are called the "zeros" of the filter, and the roots of the denominator, are called the "poles". We can factor both the numerator and denominator into their roots, and write:

$$\boldsymbol{H}(z)=\frac{z^{-(M-1)}\left(b_0 z^{M-1}+b_1 z^{M-2}+\cdots b_{M-2} z+b_{M-1}\right)}{z^{-N}\left(z^N+a_1 z^{N-1}+\cdots+a_{N-1} z^1+a_N\right)}=\frac{b_0 z^{-(M-1)}(z-z_1)(z-z_2)\cdots(z-z_{M-1})}{z^{-N}(z-p_1)\cdots(z-p_N)}$$

Notice that there are M-1 zeros, $z_{1,}\ldots,z_{M-1}$, and N poles, $p_{1,}\ldots,p_N$. At each of the zeros, $\boldsymbol{H}(z)=0$, i.e., $\boldsymbol{H}(z_k)=0$ for $k=1,\ldots,M-1$. In contrast, at each of the poles, $\boldsymbol{H}(z)$ is not defined since the denominator is zero there. Furthermore, there are z^{N-M+1} zeros (or poles, if M-1>N) at the origin $z=0$.

Example:

$$\boldsymbol{H}(z)=\frac{1+2z^{-1}+1z^{-2}}{1-z^{-1}+0.25z^{-2}}=\frac{z^{-2}(z+1)(z+1)}{z^{-2}(z-0.5)(z-0.5)}=\frac{(z+1)(z+1)}{(z-0.5)(z-0.5)}$$

This filter has two zeros at $z=-1$ and two poles at $z=0.5$

Stability from pole locations

The transfer function for the simple feedback filter

$$y[n]=a_1 y[n-1]+x[n]$$

is the function with one pole:

$$\boldsymbol{H}(z)=\frac{1}{1-a_1 z^{-1}}$$

Note that the pole is at $z=a_1$ as shown in Figure 5:

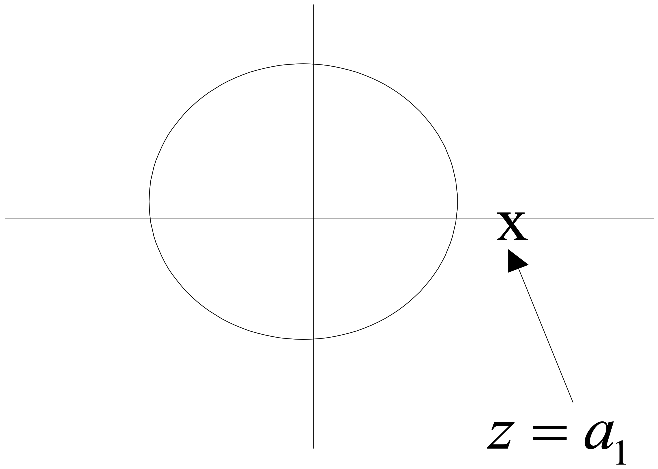

Figure 5: Pole location on the z-plane, with the unit circle $|z|=1$ *indicated.*

We've already seen by example that the filter is unstable if $|a_1|>1$. This example is a special case of the following theorem:

Stability theorem: An IIR filter is stable if and only if **all** of the poles of its transfer function $\boldsymbol{H}(z)$ lie within the unit circle, i.e., $|p_k|<1$ for all poles.

Therefore, we can determine whether a filter is stable or not by checking its pole locations. Note that the location of the zeros plays no role in stability.

For an FIR filter, the transfer function is

$$y[n]=\sum_{k=0}^{M-1} b_k x[n-k] \rightarrow \boldsymbol{H}(z)=\sum_{k=0}^{M-1} b_k z^{-k}$$

This function can be written

$$\boldsymbol{H}(z)=\frac{b_0 z^{M-1}+b_1 z^{M-2}+\cdots+b_{M-1}}{z^{M-1}}$$

This shows that the filter has M-1 zeros, and M-1 poles at $z=0$. Therefore, since all the poles are inside the unit circle, we see that *every FIR filter is stable.*

Convolution relationship

The input-output relationship of a FIR filter can be written as follows:

$$y[n]=\sum_{k=0}^{M-1} h[k]x[n-k]$$

where the impulse response $h[k]$ are the filter coefficients $h[k]=b_k$. This relationship extends even if the impulse response is not of finite-duration. Specifically, for an IIR filter

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

This is another way to obtain the output of an IIR filter, if the impulse response is known. It is also called *convolution:* we say that the output $y[n]$ is the *convolution* of the input $x[n]$ and the impulse response $h[n]$.

Network diagrams

IIR filters are often described as a network diagram. Since a delay corresponds to multiplication by z^{-1} in the z-transform, an IIR filter can be described by a network diagram as shown in Figure 6 below.

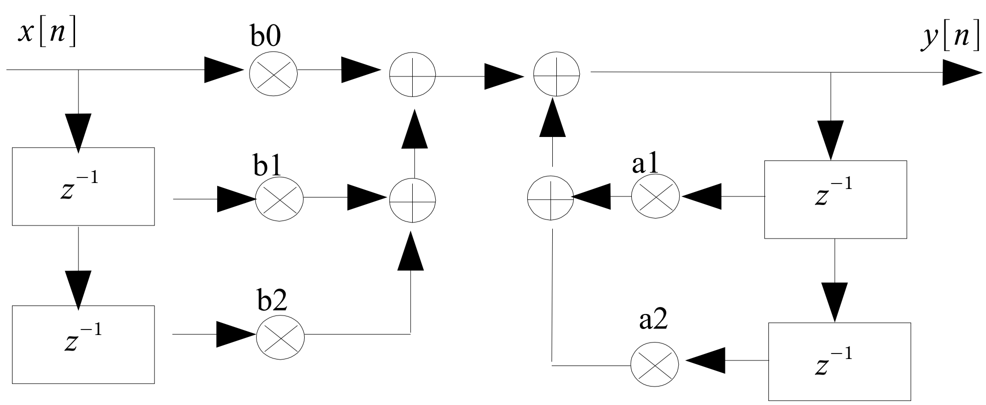

Figure 6: Network diagram for an IIR filter having two poles and two zeros.

The filter in Figure 6 is often called a second-order section (SOS).