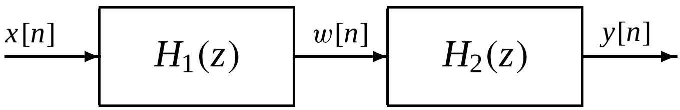
PROBLEM:

An *inverse filter* is an LTI system which, when cascaded with another LTI system, "undoes" the effects of the other LTI system. You saw an approximate inverse filter in one of the Laboratory experiments. The reason that you were only able to do an approximation then was that we had not yet studied IIR systems.



(a) Suppose that $H_1(z)$ describes a given LTI system. Determine $H_2(z)$ so that the output is $y[n] = x[n]$; i.e., so that the second system compensates exactly for the effects of the first system.

(b) If $H_1(z)$ represents an FIR system, what can you say about the second system; i.e., is it FIR or IIR? Explain.

(c) Suppose that the first system is defined by the difference equation

$$w[n] = \sum_{k=0}^{9} \alpha^k x[n-k] \qquad \text{where } 0 < \alpha < 1.$$

Show that $H_1(z)$ can be expressed the following ratio of polynomials in the variable z^{-1}:

$$H_1(z) = \frac{1 - \alpha^{10} z^{-10}}{1 - \alpha z^{-1}}.$$

Plot the poles and zeros of $H_1(z)$ in the complex z-plane. *Hint: You will need to find the values of z that satisfy the equation $1 - \alpha^{10} z^{-10} = 0$. This is done just as in the case of the moving average filter. You should find that the zeros are not on the unit circle, but on a circle of a different radius.*

(d) Now, for the system $H_1(z)$ of part (c), determine the inverse system function $H_2(z)$ and plot its poles and zeros in the complex z-plane. What happens to the poles and zeros of $H_1(z)$ and $H_2(z)$ when we form the product $H_1(z)H_2(z)$?

(e) From $H_2(z)$ obtained in part (d), determine the difference equation that relates the output $y[n]$ to $w[n]$, the input to the second system.

(f) The system of part (e) could be implemented in Matlab by the statement

```
yy=filter(b,a,ww);
```

What should `b` and `a` be in order to implement the inverse system for the example of part (c)?

McClellan, Schafer and Yoder, *Signal Processing First*, ISBN 0-13-065562-7.

SOLUTION

(a) If we want the output to be equal to the input, then for the z-transform we require $Y(z) = X(z)$. Therefore, the combined transfer function must be equal to one, $H(z) = H_1(z)H_2(z) = 1$; therefore,

$$H_2(z) = 1/H_1(z).$$

(b) If $H_1(z)$ is an FIR filter, then $H_2(z)$ is an IIR filter, because the zeros of $H_1(z)$ become the poles of $H_2(z)$,In other words, the numerator of $H_1(z)$ becomes the denominator of $H_2(z)$.

(c) First, recall summation formula for a geometric sequence: $\sum_{k=0}^{N} r^k = \frac{1 - r^{N+1}}{1 - r}$. Since the recursive difference equation for $w[n]$ is given by

$$w[n] = \sum_{k=0}^{9} \alpha^k x[n-k],$$

we can take the z-transform of the difference equation (using the shifting property) to get the system function of the first filter:

$$W(z) = \sum_{k=0}^{9} \alpha^k z^{-k} X(z)$$

$$H_1(z) = \frac{W(z)}{X(z)} = \sum_{k=0}^{9} (\alpha z^{-1})^k = \frac{1 - \alpha^{10} z^{-10}}{1 - \alpha z^{-1}}$$

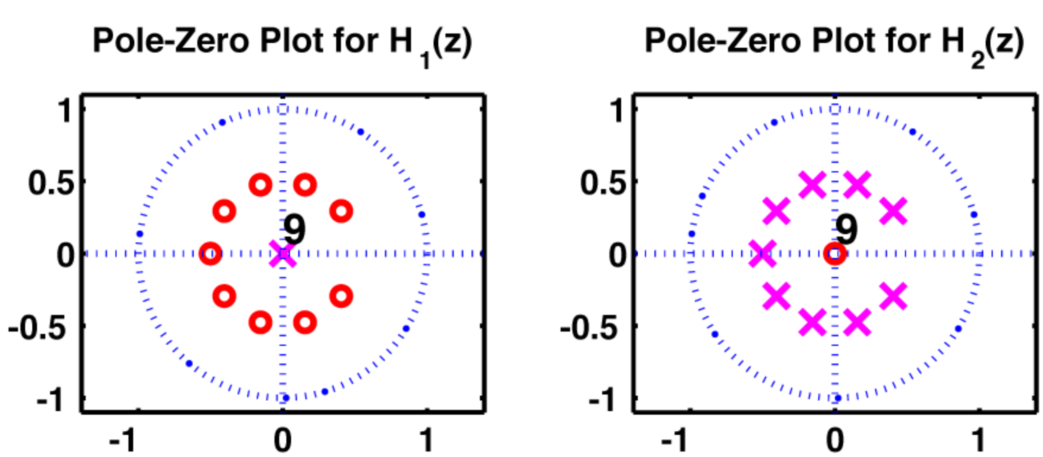
For this system function, we can write $H_1(z)$ with positive powers of z

$$H_1(z) = \frac{z^{10} - \alpha^{10}}{z^9 (z - \alpha)}$$

to see that we have 9 poles at $z = 0$, and 9 roots of α^{10} as zeros:

$$z = \alpha e^{jn\pi/10} \qquad \text{for } n = 1, 2, \ldots, 9$$

As in the case of the running average filter, there is no pole or zero at $z = \alpha$, because $H_1(\alpha)$ is not zero and it is not infinity; in fact, $H_1(\alpha)$ is equal to 10. Finally, we plot these poles and zeros below for $\alpha = 1/2$:



McClellan, Schafer, and Yoder, *Signal Processing First*, ISBN 0-13-065562-7.

This pole-zero plot can be generated with the MATLAB command called `zplane`, or the DSP-First function called `zzplane()`.

(d) There are two forms for the inverse system. The poles and zeros are plotted above.

$$H_2(z) = \frac{1}{H_1(z)} = \frac{1-\alpha z^{-1}}{1-\alpha^{10} z^{-10}}$$

$$\text{or,} \qquad H_2(z) = \frac{1}{\sum_{k=0}^{9} \alpha^k z^{-k}} = \frac{1}{1+\sum_{k=1}^{9} \alpha^k z^{-k}}$$

In the combined transfer function of $H(z) = H_1(z)H_2(z)$, for each pole we have a corresponding zero. As a result, the poles and zeros cancel each other, leaving no poles or zeros for $H(z)$, because $H(z) = 1$.

(e) There are two ways to write the difference equation for the second filter. In the first method, we use the system function:

$$H_2(z) = \frac{1}{\sum_{k=0}^{9} \alpha^k z^{-k}} = \frac{1}{1+\sum_{k=1}^{9} \alpha^k z^{-k}}$$

The denominator coefficients will be used as the feedback coefficients in the IIR filter:

$$y[n] = w[n] - \sum_{k=1}^{9} \alpha^k y[n-k]$$

In the second approach, we use the alternate form of the system function:

$$H_2(z) = \frac{1-\alpha z^{-1}}{1-\alpha^{10} z^{-10}}$$

and now when we acquire the IIR filter coefficients, we get:

$$y[n] = \alpha^{10} y[n-10] + w[n] - \alpha w[n-1]$$

It is interesting that these two systems give exactly the same output, even though they carry out different computations!

(f) As in part (e) there are two possibilities for the MATLAB code:

`bb = [1];` and `aa = [` $1\ \alpha\ \alpha^2\ \alpha^3\ \alpha^4\ \alpha^5\ \alpha^6\ \alpha^7\ \alpha^8\ \alpha^9$ `]`

or, `bb = [1,`$-\alpha$`];` and `aa = [1, zeros(1,9),` $-\alpha^{10}$`]`

In either case, the call to the `filter()` function is: `yy = filter( bb, aa, xx );`

McClellan, Schafer, and Yoder, *Signal Processing First*, ISBN 0-13-065562-7.