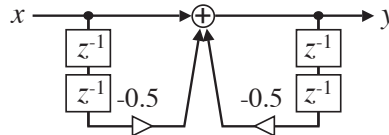


1. (a) Rearranging to get  $y[n]$  on the left-hand side gives:

$$y[n] = x[n] - 0.5x[n - 2] - 0.5y[n - 2] \tag{1}$$

A simple block diagram implementation would be:

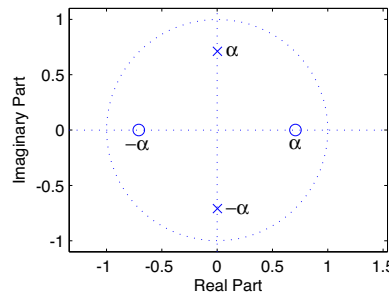


(although there are many variants that could eliminate delays and/or multiplies).

- (b) Factorizing the z-transform, we find a pair of real zeros, and a pair of imaginary poles, all at radius  $\alpha = \sqrt{0.5} \approx 0.707$ :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - 0.5z^{-2}}{1 + 0.5z^{-2}} = \frac{(1 + \alpha z^{-1})(1 - \alpha z^{-1})}{(1 + j\alpha z^{-1})(1 - j\alpha z^{-1})} \tag{2}$$

with region of convergence  $|z| > \alpha$ . Thus, the pole-zero diagram is:



- (c) The input consists of two sinusoids, one with frequency zero (i.e. a constant value), one with frequency  $\pi/2$  rad/samp. We are given no time limits, so we can treat these as infinite-duration sinusoids. The output will thus be the same sinusoids with magnitude and phase modified by the magnitude and phase response of the system at those frequencies. We can get these graphically, or simply by evaluating the system function  $H(z)$  at  $z = e^{j\omega}$  for the corresponding frequencies. Notice that  $H(z)$  is real in both cases, meaning that both components experience zero phase shift. For  $\omega = 0$ ,  $H(e^{j\omega}) = H(1) = (1 - 0.5)/(1 + 0.5) = 1/3$ . For  $\omega = \pi/2$ ,  $H(e^{j\omega}) = H(j) = (1 + 0.5)/(1 - 0.5) = 3$ . Thus, the system output  $y[n] = 1/3 + 3 \cos(\omega_0 n)$ .
- (d) Substituting  $jz$  for  $z$  i.e.  $-jz^{-1}$  for  $z^{-1}$ , we get:

$$G(z) = \frac{1 - 0.5j^2z^{-2}}{1 + 0.5j^2z^{-2}} = \frac{1 + 0.5z^{-2}}{1 - 0.5z^{-2}} \tag{3}$$

(We note in passing that, for this particular system, this transformation gives us the inverse system). The region of convergence remains  $|z| > \alpha$ .

At  $z = 1, 2$ , we can simply evaluate our  $G(z)$ :

$$G(1) = (1 + 0.5)/(1 - 0.5) = 3 \quad (4)$$

$$G(2) = (1 + 0.5/4)/(1 - 0.5/4) = (9/8)/(7/8) = 9/7 \quad (5)$$

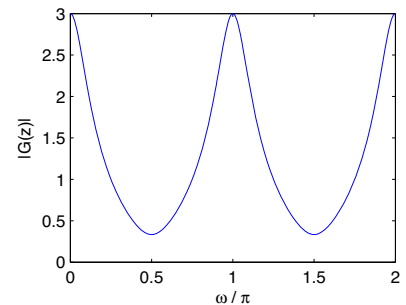
For  $z = 0, 0.5$ , however, we are not in the region of convergence. The correct response is that  $G(z)$  does not converge at these points.

- (e)  $G(z)$  has two poles on the real axis, and two zeros on the imaginary axis (it is 90 deg rotated from the pole-zero diagram of part (b)). From graphical considerations, we expect it to have maxima at  $\omega = 0, \pi, 2\pi$  and minima mid-way between the maxima at  $\omega = \pi/2, 3\pi/2$ . Substituting  $z = e^{j\omega}$  into eqn. 3, we get:

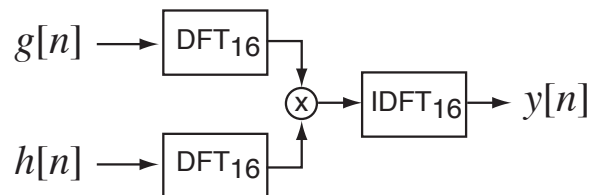
$$G(e^{j\omega}) = \frac{1 + 0.5e^{-2j\omega}}{1 - 0.5e^{-2j\omega}} \quad (6)$$

This isn't going to resolve into anything simple like a single sinusoid, so we can just sketch a curve with the appropriate maxima and minima. We already calculated  $H(z)$  for  $z = 1, j$  in part (c), so we have some exact values we can label. In fact, the peaks are "skinnier" than the poles (when plotted on a linear magnitude axis):

```
>> [G,W] = freqz([1 0 .5],[1 0 -.5],[0:.01:2*pi]);
>> plot(W/pi,abs(G))
>> xlabel('omega / pi');
>> ylabel('|G(z)|')
```



2. (a) This is straightforward convolution by DFT/IDFT. The block diagram I intended is: That is, we



pad both sequences with zeros to extend them both to 16 points, then calculate 16 point DFTs of each,  $H[k]$  and  $G[k]$  for  $k = 0 \dots 15$ , multiply these two together to get  $Y[k] = H[k]G[k]$ , then perform the 16 point IDFT to get the 16 point result,  $y[n]$  for  $n = 0 \dots 15$ .

The linear convolution of a 12 point sequence with a 5 point sequence will have  $12 + 5 - 1 = 16$  points, hence the use of 16 point DFTs. Of course, we could use a longer DFT and still get the correct result, but at greater computational expense. A shorter DFT will result in time aliasing.

- (b) Since, by construction, our system will calculate linear convolution, the answer here is simply  $h[n] \otimes g[n]$ .  $g[n]$  consists of a delta, then a second, negated delta delayed by four samples. Thus,  $y[n] = h[n] - h[n - 4]$ . By inspection, we can write this down as  $\{1, 2, 3, 4, 4, 4, 2, 0, -2, -4, -4, -4, -4, -3, -2, -1, 0\}$ .

- (c) This is close to the same computation, but with two exceptions: (a) we are using 8-point DFTs, so we are going to get an 8 point result, with the inevitable time aliasing; and (b) the operation that ought to be the final IDFT is in fact just another DFT. We know that DFT and IDFT are similar, but whereas the IDFT of  $Y[k]$  is (by definition)  $y[n]$ , the DFT applied to the same sequence gives a scaled and circular-time-reversed version – if  $\text{IDFT}\{Y[k]\} = y[n]$ , then  $\text{DFT}\{Y[n]\} = Ny[\langle -k \rangle_N]$  (topic 3, slide 55).

Thus, starting with the answer to part (b), we first time-alias down to 8 points by adding the second half to the corresponding points in the first half:

$$\{1 - 2, 2 - 4, 3 - 4, 4 - 4, 4 - 3, 4 - 2, 2 - 1, 0 - 0\} = \{-1, -2, -1, 0, 1, 2, 1, 0\} \quad (7)$$

This would be the output with an IDFT. To get the result from DFT, we circularly-time-reverse (meaning we keep the zero point in-place and flip the rest), and scale everything by  $N$  (8, in this case). I.e.:

$$8 \times \{-1, \text{flipr}\{-2, -1, 0, 1, 2, 1, 0\}\} \quad (8)$$

$$= \{-8, 0, 8, 16, 8, 0, -8, -16\} \quad (9)$$

Let's just check that in Matlab, to be sure. Note that we have to calculate the 16 point DFT of  $h$ , then subsample it by 2, because Matlab will simply truncate (instead of time aliasing) if we try to take the 8-point DFT of a 12-point sequence.

```
>> H = fft(h, 16);
>> H = H(1:2:end);
>> G = fft(g, 8);
>> fft(G.*H)
ans =
-8.0000 0 8.0000 16.0000 8.0000 0 -8.0000 -16.0000
```