The output response y[n] of the system was calculated using the inverse Discrete Time Fourier Transform (DTFT) of the product of the DTFT’s of the input signal x[n] and the transfer function h[n].

In order to calculate the DTFT of the input signal, x[n] was represented as three sinusoids. The constant component was represented as a cosine wave with a frequency of 0, while the sinusoidal inputs were represented as sine waves with frequencies of pi/2 and pi/4. The DTFT of each of these signals was calculated using the appropriate DTFT pairs in Table 4.1 (pg. 177 of Kamen and Heck).

The DTFT of the transfer function was also calculated a DTFT pair in Table 4.1 and multiplied with the DTFT’s of the input signals. Additionally, the amplitude and phase responses of the transfer function were calculated for the existing input frequencies. This allowed for the product of the transfer functions’ DTFT’s and the input functions’ DTFT’s to be represented in polar form.

After the product of the of the input functions’ and output functions’ DTFT’s was calculated to represent the output response Y(omega) of the system in the frequency domain, the inverse DTFT of the output response was calculated using DTFT pairs in Table 4.1. This yields the objective function y[n] which is the time domain representation of the system’s output for the given input signal x[n].

To verify the results, the calculated output function was plotted using MATLAB and compared to a plot of the convolution of the time domain representations of input function and transfer function. This is valid since convolution in the time domain is equivalent to multiplication in the frequency domain.

Here is the MATLAB used to carry out this process:

>> syms x y Y h n

>> n = (0:1:63);

>> x = 1 + sin(pi\*n/2) + sin(pi\*n/4);

>> h = 1.9\*(-0.9).^n;

>> Y = conv(x, h);

>> plot(Y);

>> n = (0:1:127);

>> y = 1 + 1.41\*sin(n\*pi/2) + 1.08\*sin(n\*pi/4+0.371);

>> plot(y);

The plots of the convolution function Y and the analytically calculated function y are shown as figures 1 and 2, respectively. It should be noted that the decaying sinusoid at the end of figure 1 results from the “window” of convolution operation passing along the decaying exponential of h[n], while this decay isn’t seen in figure 2 because this function is directly composed of sinusoids.

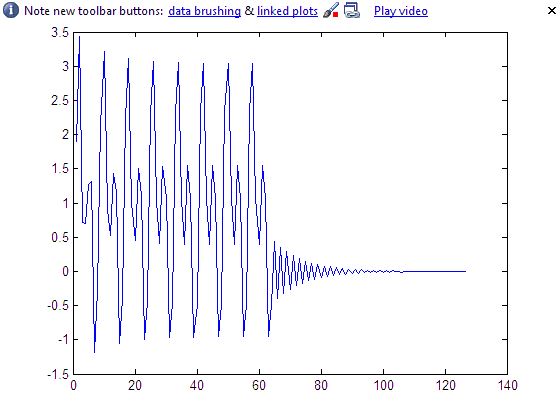


Figure 1: Convolution of input and transfer function (-1 < n < 64)

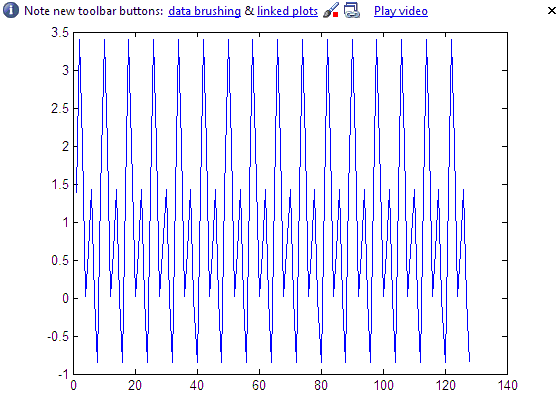


Figure 2: Plot of analytically calculated output (-1 < n < 128)

\*\*\*\* I’ve worked this problem by hand (rather neatly) with an explanation of the steps being taken and the calculations. I plan to scan this solution and add it to this document once I have access to a scanner. I’ll resubmit this document and give you the solutions after class, but I assumed it would be wise to turn in what of my work I can before the given due date.