Mark Williams

Signals and Systems

03/27/2009

MARK WILLIAMS

→ GIVEN: $h[n] = 1.9(-0.9)^n u[n]$, $x[n] = 1 + \sin(\frac{\pi}{4}n) + \sin(\frac{\pi}{2}n)$

1.) THE IMPULSE FUNCTION IS FOUND BY: $h[n+1] + .9h[n] = 1.9\delta[n+1]$

$h[n+1] + .9h[n] = 1.9\delta[n+1]$

$1.9(-.9)^{n+1}u[n+1] + .9(1.9)(-.9)^n u[n]$

$= 1.9(-.9)^{n+1}(u[n+1] - u[n])$

$= 1.9\delta[n+1]$

BY SUBSTITUTING $h[n] = 1.9(-.9)^n u[n]$

FACTOR OUT $1.9(-.9)^{n+1}$

B = 0 FOR ALL EXCEPT $n = -1$, IF $n = -1$ THEN $(-.9)^{n+1} \to 1$

$\Rightarrow 1.9\delta[n+1]$

$\{ = u[n+1] - u[n]$

$= \delta[n+1]$

THE FREQUENCY RESPONSE FUNCTION MUST THEN BE FOUND:

$h[n] = 1.9(-.9)^n u[n]$, THEN

$$H(\omega) = \frac{1.9}{1 + .9e^{-j\omega}} = \frac{1.9e^{j\omega}}{e^{j\omega} + .9}$$

→ $X(\omega) = 2\pi\delta(\omega) + j\pi[\delta(\omega + \pi/4) - \delta(\omega - \pi/4)] + j\pi[\delta(\omega + \pi/2) - \delta(\omega - \frac{\pi}{2})]$

$Y(\omega) = H(\omega)X(\omega)$

$$H(0) = \frac{1.9}{1 + .9} = 1 \quad H(\pi/4) = \frac{1.9e^{j\pi/4}}{e^{j\pi/4} + .9} \quad H(\pi/2) = \frac{1.9e^{j\pi/2}}{e^{j\pi/2} + .9}$$

$y[n] = 1 + 1.08\cos(\pi/4\, n + \angle H(\pi/4)) + 1.41\cos(\pi/2\, n + \angle H(\pi/2))$

$$H(\pi/4) = \frac{1.9(\cos\pi/4 + j\sin\pi/4)}{\cos\pi/4 + j\sin\pi/4 + .9} = \frac{1.34 + j1.34}{1.6 + j.707} \Rightarrow |H(\omega)| = 1.08 \Rightarrow \angle H(\pi/4) = .37\text{RAD}$$

$$H(\pi/2) = \frac{1.9(\cos\pi/2 + j\sin\pi/2)}{\cos\pi/2 + j\sin\pi/2 + .9} = \frac{1.9(0 + j(1))}{.9 + j \cdot 1} = \frac{j1.9}{.9 + j} \Rightarrow |H(\pi/2)| = 1.41$$

$\angle H(\pi/2) = .733\text{RAD}$

$$\boxed{y[n] = 1 + 1.08\sin(\pi/4\, n + .37\text{RAD}) + 1.41\sin(\pi/2\, n + .733\text{RAD})}$$

MY WORK

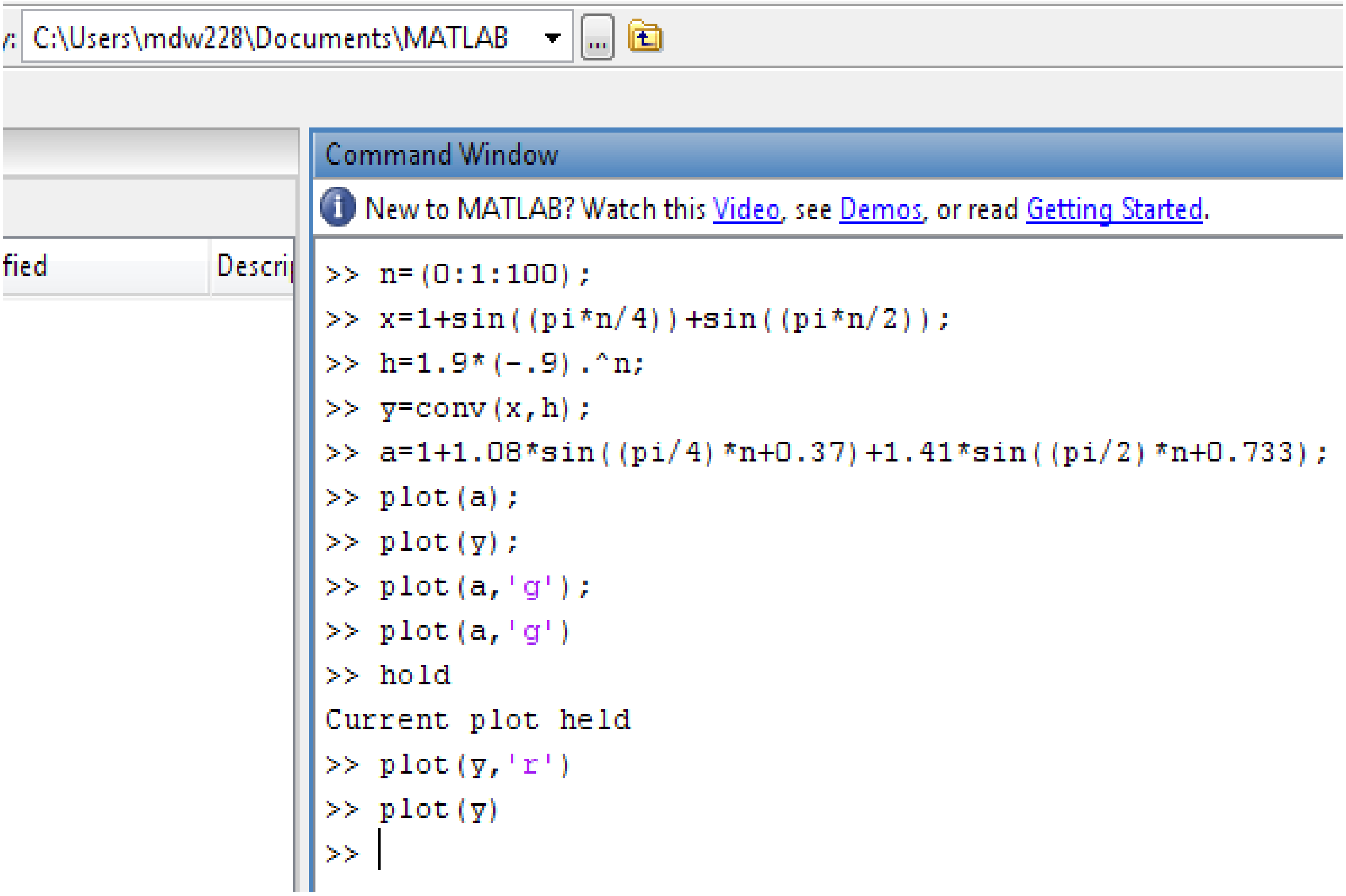

Figure-1 Matlab code

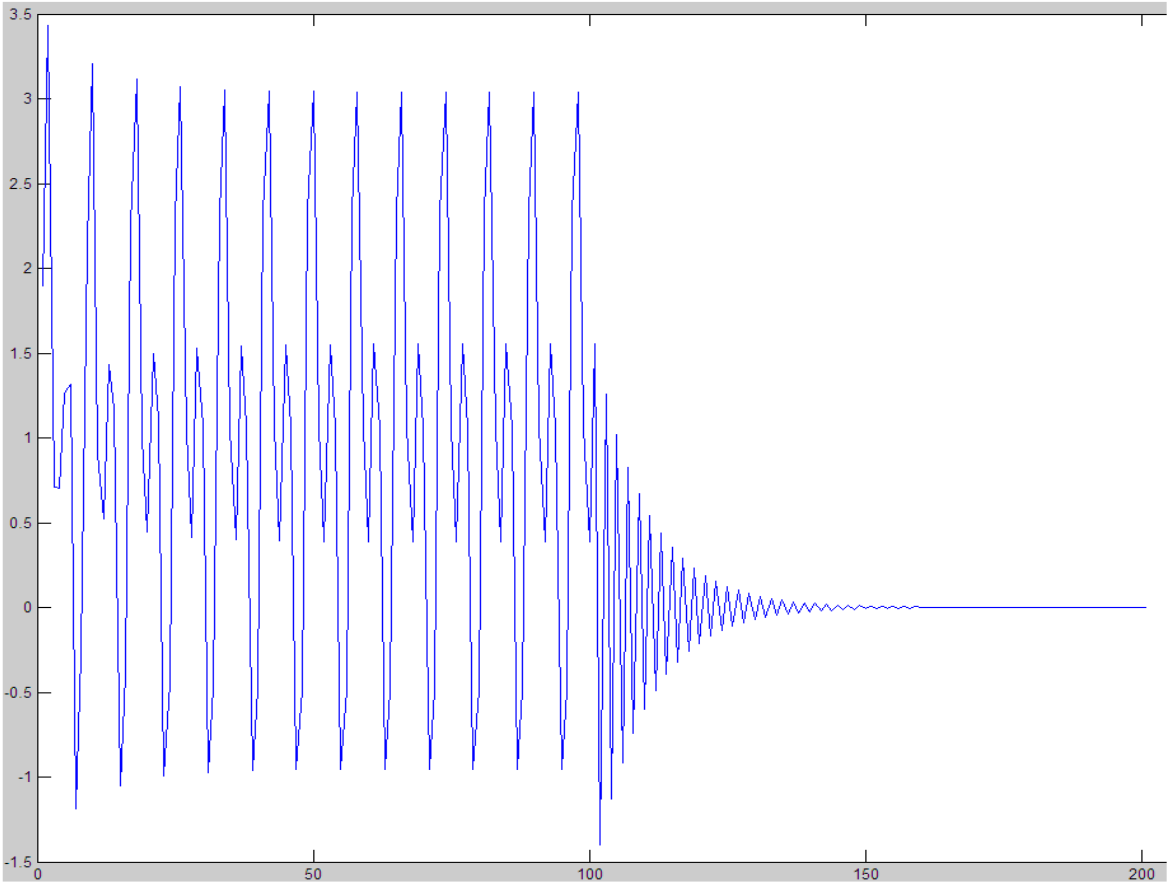

Figure-2 Matlab solution

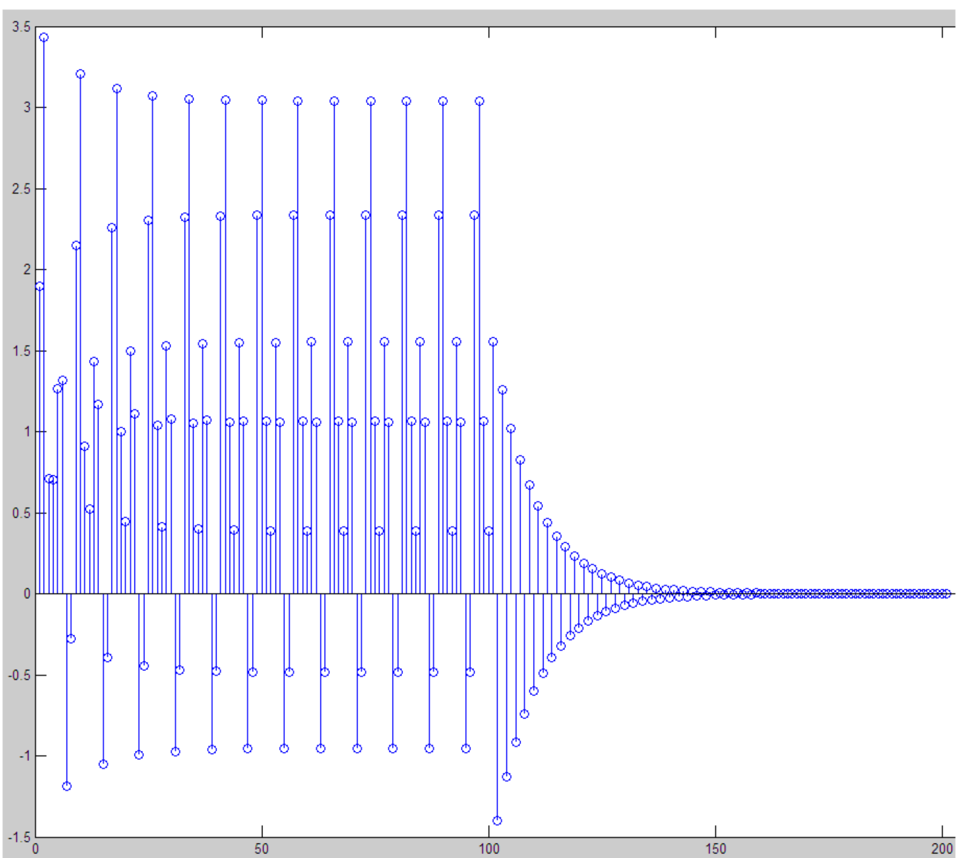

Figure-3 Matlab solution in ‘stem’ plot - I just thought this one looked cool

```
>> x=1+sin((pi*m/4))+sin((pi*m/2));
>> h=1.9*(-.9).^m;
>> z=conv(x,h);
>> plot(z)
```

Figure – 4 – vector set to 300 code

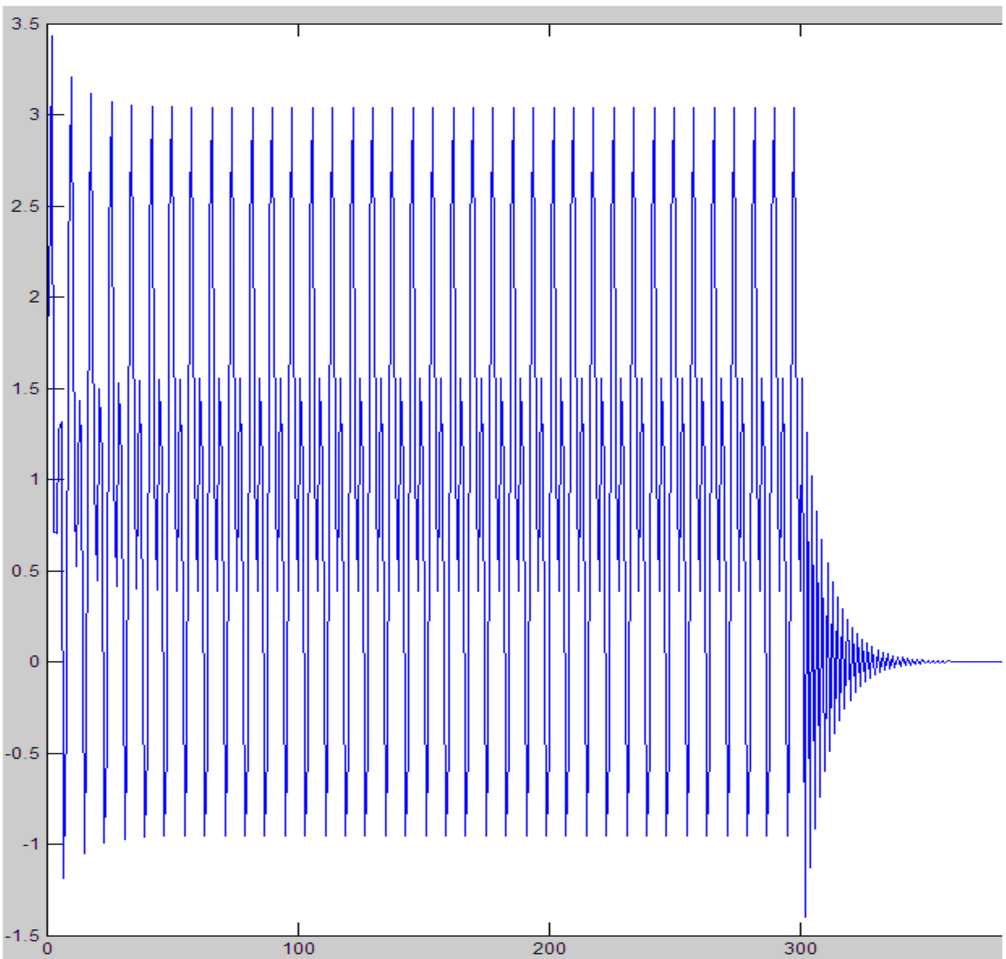

Figure-5- vector set to 300 plot

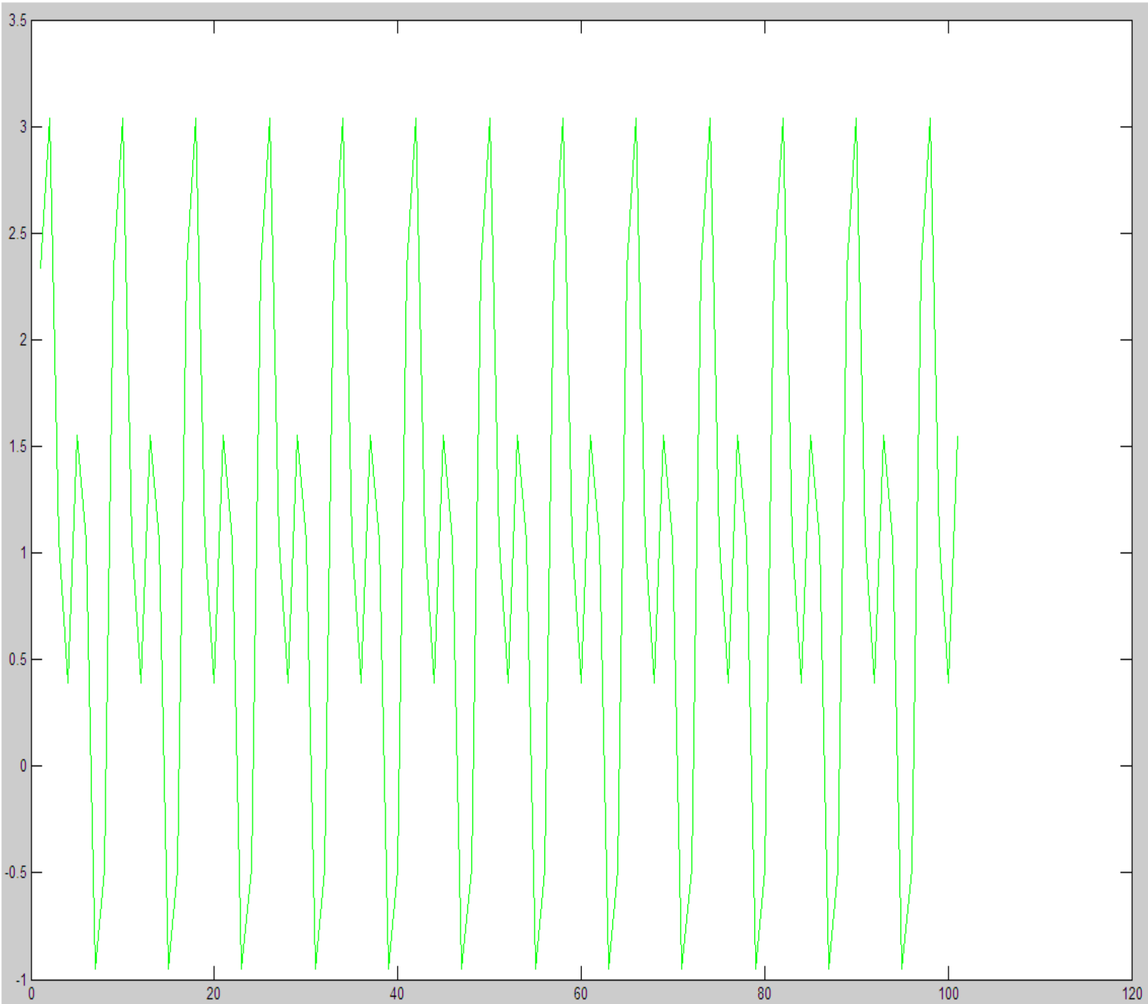

Figure-6   Analytical Solution

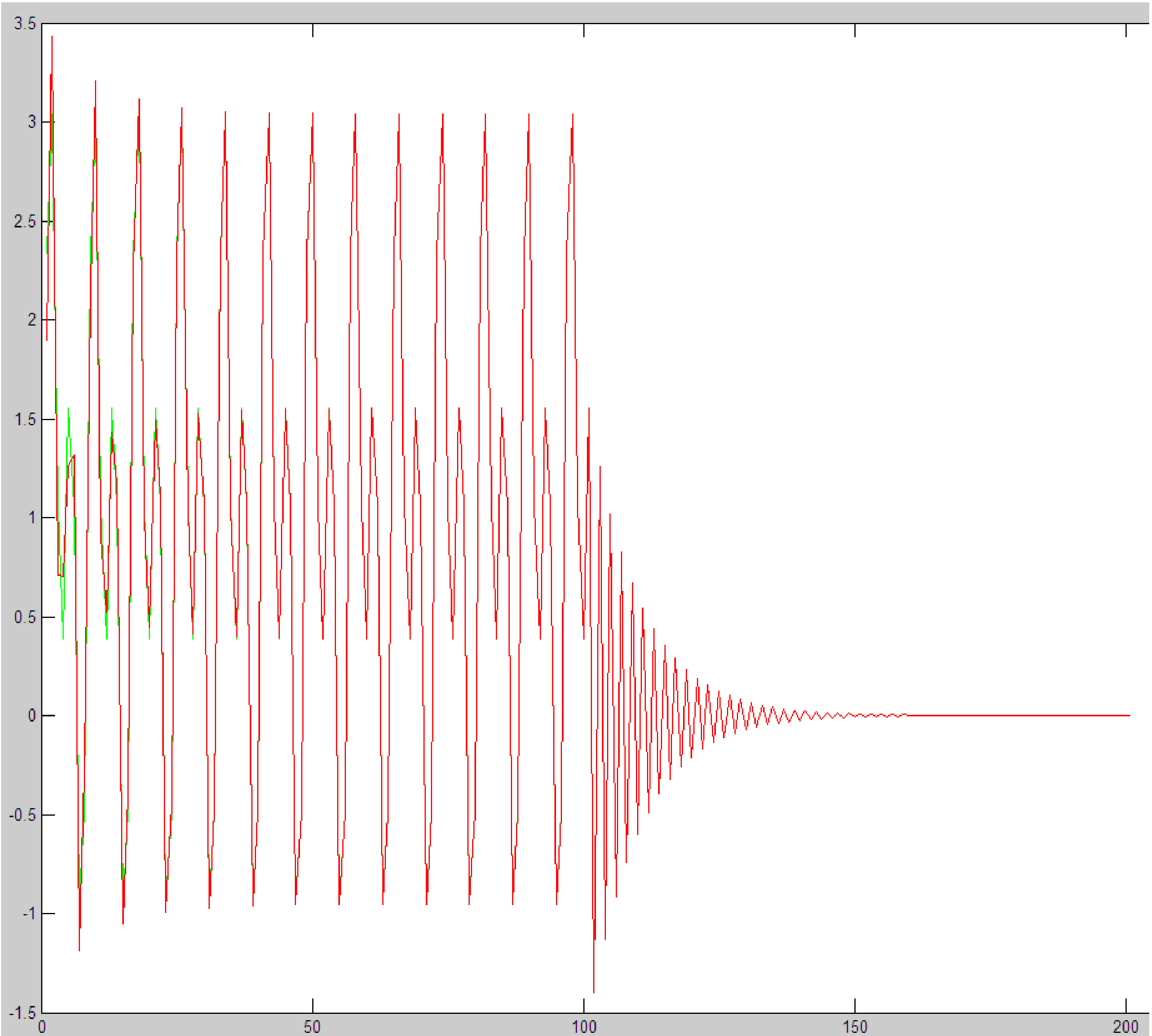

Figure-7 Analytical and Matlab solution plotted together

Explanation of Matlab Simulation

The output, y[n] was found by doing the convolution of the input signal; x[n]and the unit pulse response function h[n]. A vector, 'n' was designated to give a sufficient amount of room to view the convolution. I tried two different values for the vector; just help me understand the result better. In Figure 2, Figure 3, and Figure 5 there is a plot of my Matlab solution. As my results show, y[n] is a continuous signal that we are showing as far as we set our vector to show. The decay of the signal is approximately 40 in both cases. In figure 6, my analytical plot is displayed. Once again the only difference between the simulation and the analytical is that the analytical shows the continuous nature of the output, thus leaving out the decay. In figure 7 I have plotted both my analytical and simulated values. The two signals line up almost perfectly (Obviously they can't be EXACTLY the same). This simulation made me appreciate Matlab a tremendous amount more. It is a powerful tool, when used correctly.

Explanation of Analytical work

From my understanding of the analytical work, in order to find the output, a convolution must be taken between the input and the unit pulse response. Since convolution in the time domain is the same as multiplication in the frequency domain, a quick conversion to a DTFT makes our effort possible. Also, h[n] is converted into a frequency response function. The following equation sums it all up.

$$Y(\Omega)=X(\Omega)H(\Omega) \qquad (1)$$

From there I calculated the output's, magnitude and angle at the different input frequencies given. The frequencies were plugged into the frequency response function calculated earlier in the problem. Through, several intense calculations my magnitudes and phase angles are combined for my final output function.