③

Good work!

Dan Ports
6.003 PS4

1.

3.34 a. $x(t) = \sum_{n=-\infty}^{\infty} \delta(t-n) \Rightarrow T = 1$

$a_0 = 1$

$a_k = \frac{1}{T}\int_{-1/2}^{1/2} \delta(t) e^{-j2\pi k} \, dt = 1$

So all fourier coefficients of the input are 1.

The transfer function $H(j\omega) = \int_{-\infty}^{\infty} h(t) e^{-j\omega t}$

$= \int_{-\infty}^{\infty} e^{-4|t|} e^{-j\omega t} = \int_{-\infty}^{0} e^{(4-j\omega)t} + \int_{0}^{\infty} e^{(-4-j\omega)t}$

$= \frac{1}{4-j\omega} e^{(4-j\omega)t}\Big|_{-\infty}^{0} + \frac{1}{-4-j\omega} e^{(-4-j\omega)t}\Big|_{0}^{\infty}$

$= \frac{1}{4-j\omega} + \frac{1}{j\omega+4} = \frac{8}{\omega^2+16}$

So the output Fourier coefficients will be

$b_k = H(jk2\pi/T)\, a_k = \frac{8}{(2\pi k)^2+16} = \boxed{\frac{2}{(\pi k)^2+4} \quad \forall k}$ ✓

good clean math

c. $T = 1$.

$a_0 = 1/2$

$a_k = \int_{-1/4}^{1/4} e^{-j2\pi k t} \, dt = \frac{1}{-j2\pi k t} e^{-j2\pi k t}\Big|_{-1/4}^{1/4}$

$= \frac{e^{j\pi/2 \, kt} - e^{-j\pi/2 \, kt}}{2\pi j k} = \frac{\sin k\pi/2}{\pi k}$

So

$b_k = H(jk2\pi) a_k = \begin{cases} \frac{8}{16}\cdot\frac{1}{2} = \frac{1}{4}, & k = 0 \\ \frac{8}{(2\pi k)^2+16}\cdot\frac{\sin k\pi/2}{\pi k}, & k \neq 0 \end{cases}$

5

Good work!

3.39a With period $N=3$, $\omega_0 = 2\pi/3$.

Then each period has Fourier series coefficients a_0, a_1, a_2.

For a_0, $\omega = 0 \Rightarrow H(e^{j\omega}) = 1$

a_1, $\omega = 2\pi/3 \Rightarrow H(e^{j\omega}) = 0$

a_2, $\omega = 4\pi/3 \Rightarrow H(e^{j\omega}) = H(e^{j\cdot -2\pi/3}) = 0$.

So the output coefficients b_1 and b_2 will be zero; b_0 is the only non-zero one.

3.60 g, h, i

g. $$x(n) = \frac{e^{j\pi n/3} + e^{-j\pi n/3}}{2}$$

$$y(n) = \frac{e^{j\pi n/3} + e^{-j\pi n/3}}{2} - \sqrt{3}\,\frac{e^{j\pi n/3} - e^{-j\pi n/3}}{2j}$$

So there is more than one LTI system with this input/output pair! It simply needs $H(e^{j\pi/3}) = 1 - j\sqrt{3}$!

and $H(e^{-j\pi/3}) = 1 + j\sqrt{3}$

please write more visibly

h. $y[n]$ is two shifted copies of $x(n)$, so we can use the LTI system with impulse response $\delta(n-3) + \delta(n+3)$. But this is not unique; we could also use $\delta(n-13+k_1 12) + \delta(n-13+k_2 12)$ for $k_1, k_2 \in \mathbb{Z}$

i. $y[n]$ has period 9 and $x(n)$ has period 12. There is no LTI system that can perform this transform.

3.64.

a. $y(t)$ is simply $\sum_{k=-\infty}^{\infty} \lambda_k c_k \varphi_k(t)$.

b. This is a linear ODE, but it does not have constant coefficients. So it is linear but not TI.

c. $\varphi_k(t) = t^k$

$$t^2 \frac{d^2\varphi_k(t)}{dt^2} + t\frac{d\varphi_k(t)}{dt} = t^2 k(k-1)t^{k-2} + t\,k\,t^{k-1}$$

$$= (k^2 - k + k)t^k = k^2 t^k = \lambda_k \varphi_k(t)$$

for $\lambda_k = k^2$

d. $x(t) = 10\,\varphi_{-10}(t) + 3\,\varphi_1(t) + 1/2\,\varphi_4(t) + \pi\,\varphi_0(t)$

$\rightarrow y(t) = (-10)^2 \cdot 10\,\varphi_{-10}(t) + 1^2 \cdot 3\,\varphi_1(t) + (4)^2 \cdot 1/2\,\varphi_4(t) + 0 \cdot \pi$

$= 1000\,\varphi_{-10}(t) + 3\varphi_1(t) + 8\,\varphi_4(t)$

$= 1000\,t^{-10} + 3t + 8t^4$

5a. See attached plots. why are the coeff's large for hi, lo indices?

b. We can make the Fourier series coefficients real by making the signal even:
$x[-n] = x[100-n] = x[n]$
Thus $x[50 \ldots 99] = [50 \ldots 2]$. Plot attached.

c. The sequence $x[n]$ is real and even, so its Fourier coefficient are real and even. So the sequence can be written as

$$\sum_{k=0}^{49} a_k e^{jwk} + a_k e^{-jwk} = \sum_{k=0}^{49} a_k \left(e^{jwk} + e^{-jwk}\right)$$

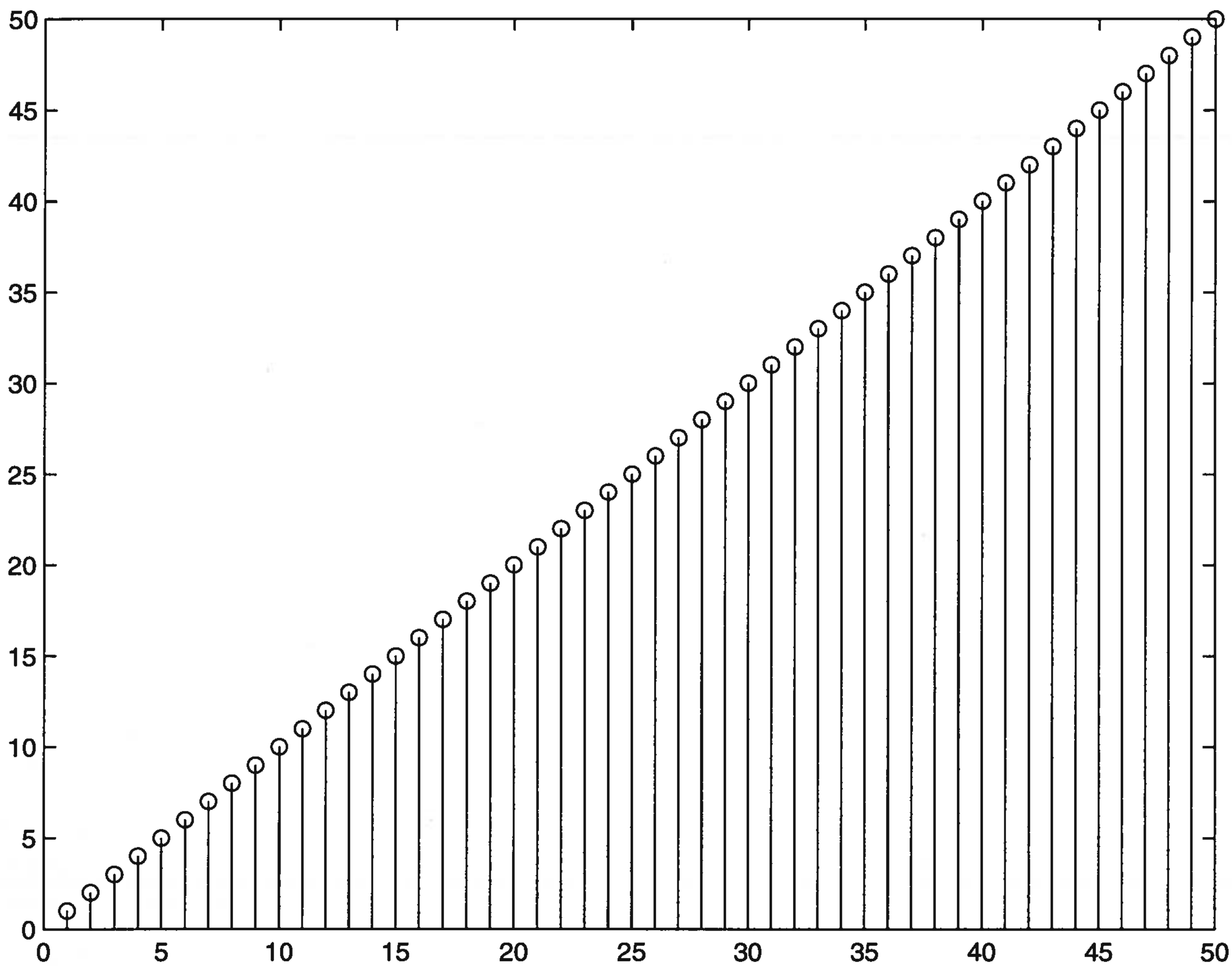
$$= \sum_{k=0}^{49} a_k \cos\left(\frac{2\pi kn}{N}\right) \quad \text{for} \quad a_k = \frac{b_k}{2}$$

d. See attached plot.

e. The Fourier coefficients shown in the new plot, besides being real, are also even. This means half of them do not need to be stored, since we can reconstruct them from the other half. Also, many of the coefficients are zero. A data compression algorithm could ignore them and only store the other coefficients.

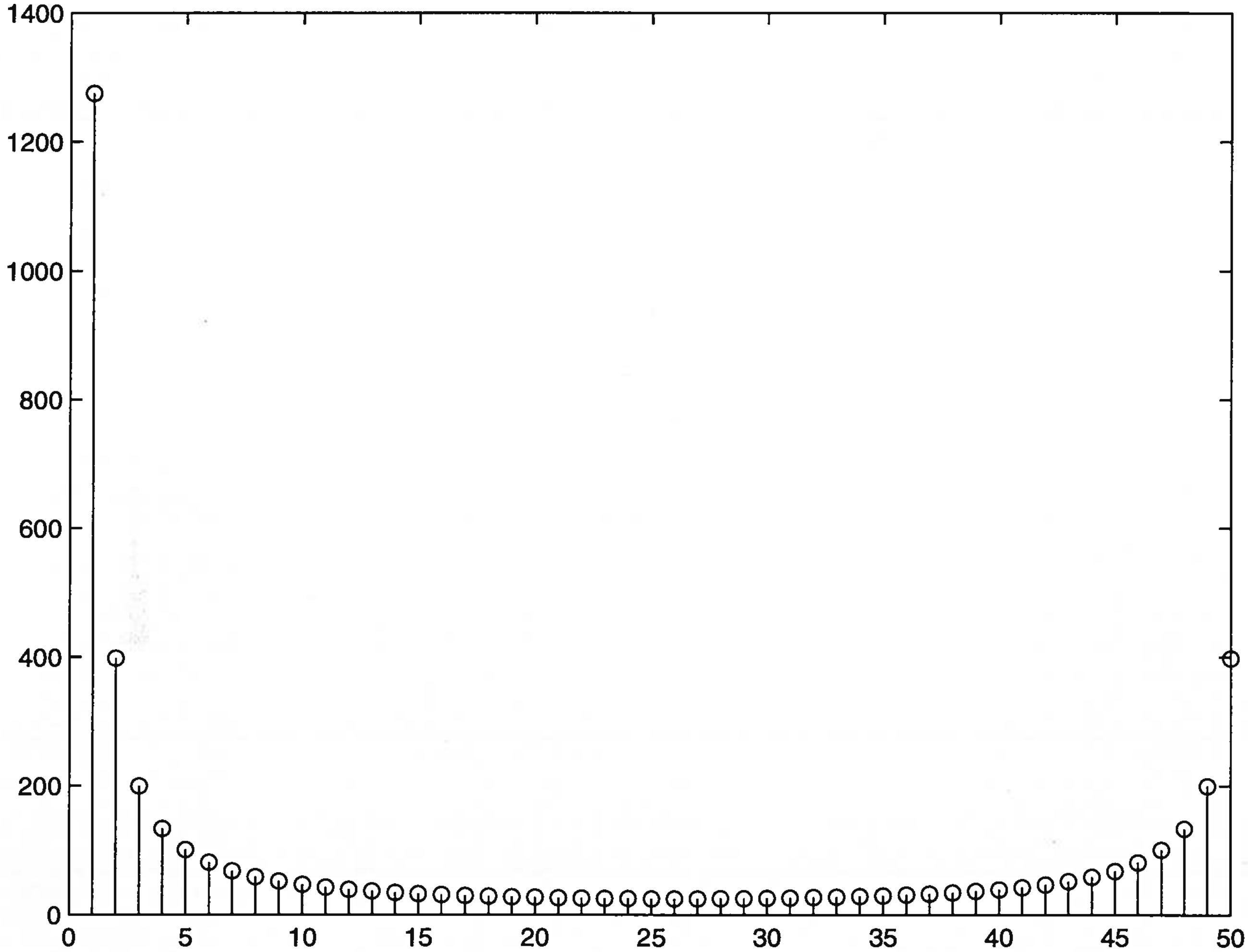
(5a)

stem(x)



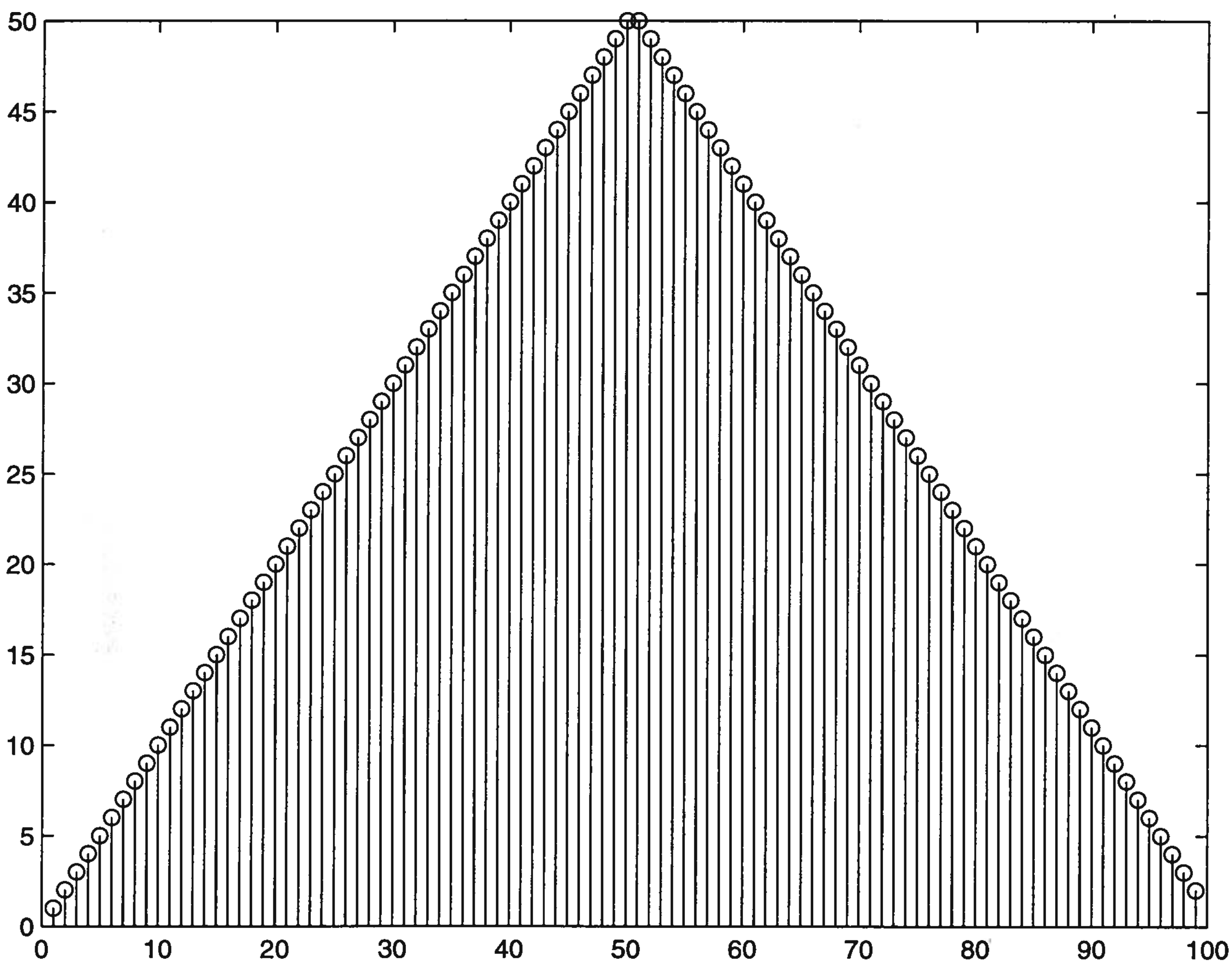
. (5a)

stem(abs(fft(x)))



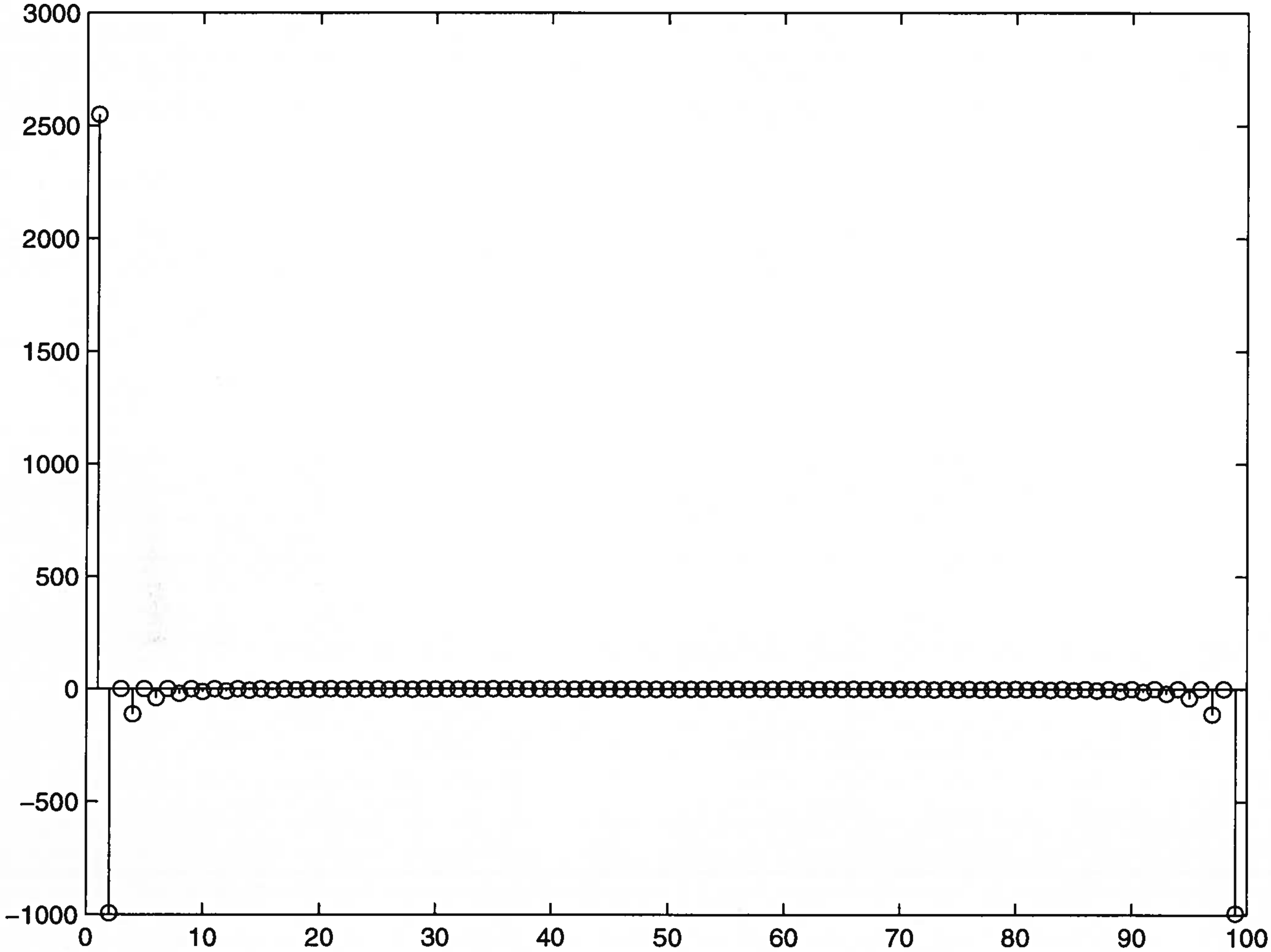
. 5b.)

Stem(v) after augmenting v



(5d)

stem(real(fft(x))) after augmenting x.



6. We note that $H_1(e^{jw}) = \begin{cases} 20, & w=0 \\ \frac{1}{1.45}, & w=\pi \end{cases}$

and $H_2(e^{jw}) = \begin{cases} 0, & w=0 \\ -2 & w=\pi \end{cases}$

So H_1 is a low-pass filter and H_2 is a high-pass filter

Image 2 has the detail removed from the rows, so filter H_1 was applied to the rows. Similarly, image 3 had filter H_1 applied to the columns.

Image 4 contains the details of each row but the average color is gray; it had a high-pass filter applied to the rows, so H_2 was applied to the rows. Similarly, H_2 was applied to the columns in image 5.

7. a Stem plot attached. We see a sinusoid signal that satisfies (approximately) hum[x] ≈ hum[x+20] ⇒ $\omega = \frac{2\pi}{20}$ ✓

Of course, it is not a pure sinusoid! It is the combination of the sinusoid w/ $\omega = 2\pi/20$ that represents the 400 Hz hum combined with a non-periodic signal that is the rest of the audio.

b. Applying the firnotch filter makes the audio intelligible, but it is noticeably ... by the attached ... the hum at frequency $\pi/10$ is essentially eliminated but the notch is not very narrow and other frequencies less than $\pi/5$ are also attenuated somewhat. ✓

c. This equation gives a notch filter. Increasing A makes the notch sharper, but not as deep. Decreasing B increases the frequency that is notched out, and vice versa. Increasing B also requires A to be decreased to maintain the same level of attenuation. ✓

d. Tried various values of B before finding that B = 1.9 centers the notch at $\pi/10$. Choice of A is more difficult; if A is too large, the notch does not provide enough attenuation, if A is too small, the notch is too wide. ✓

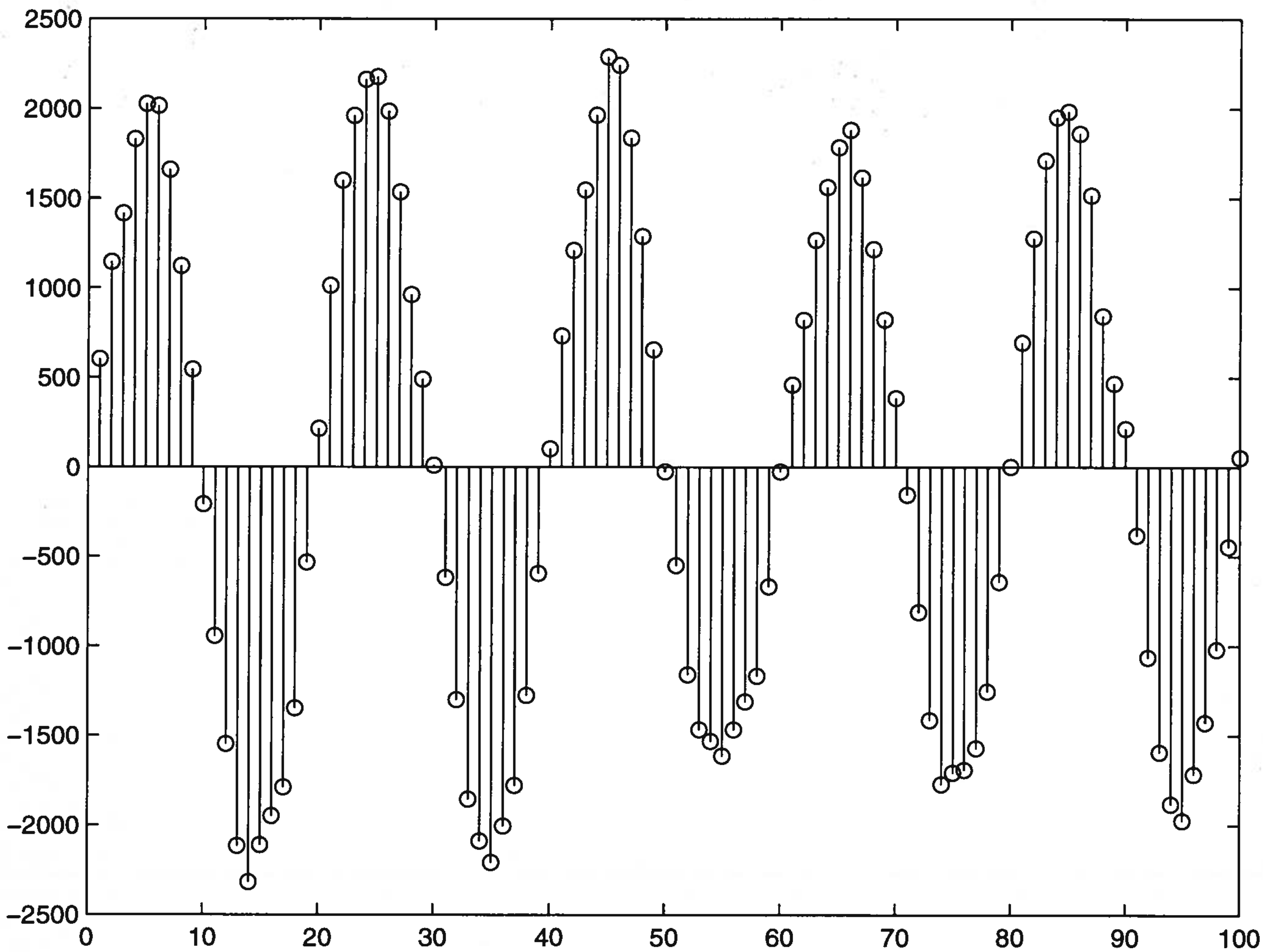
A = .85 seems a good compromise; it provides about 40 dB attenuation and is fairly narrow. With this filter, the hum is only slightly audible, but the neighboring frequencies are not overly affected. Frequency plot attached.

e. The IIR filter can produce a much better notch filter than the FIR filter because the FIR filter can only take into account 20 points of input for each output point; but the IIR filter can take into account every input point. This also makes the IIR filter ? more computationally expensive.

FIR needs 20 values to compute the next y[n]; the filter in (c) only needs 5...

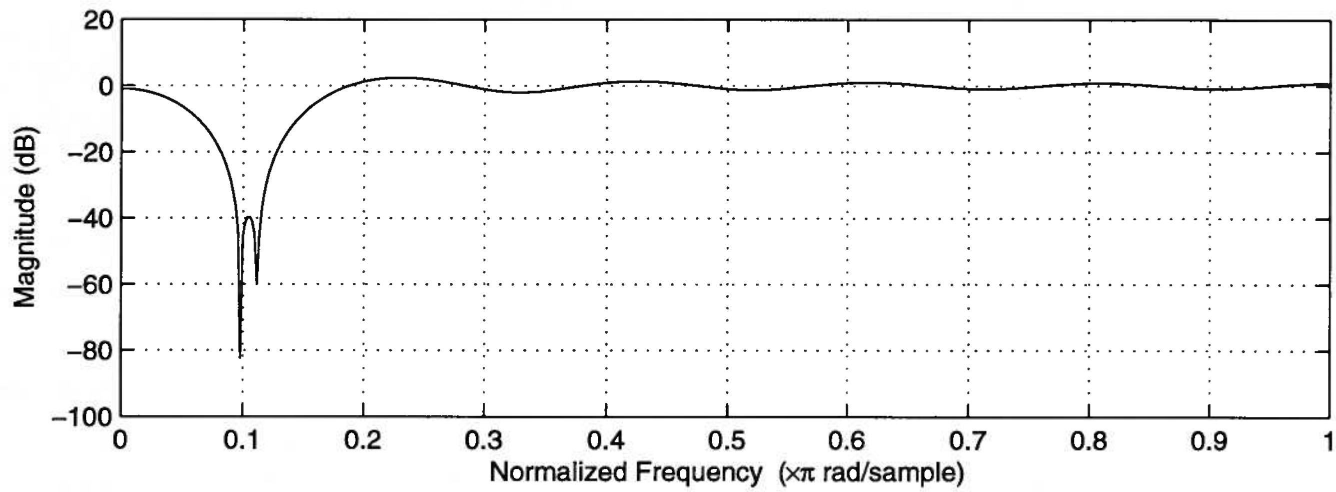
(7a)

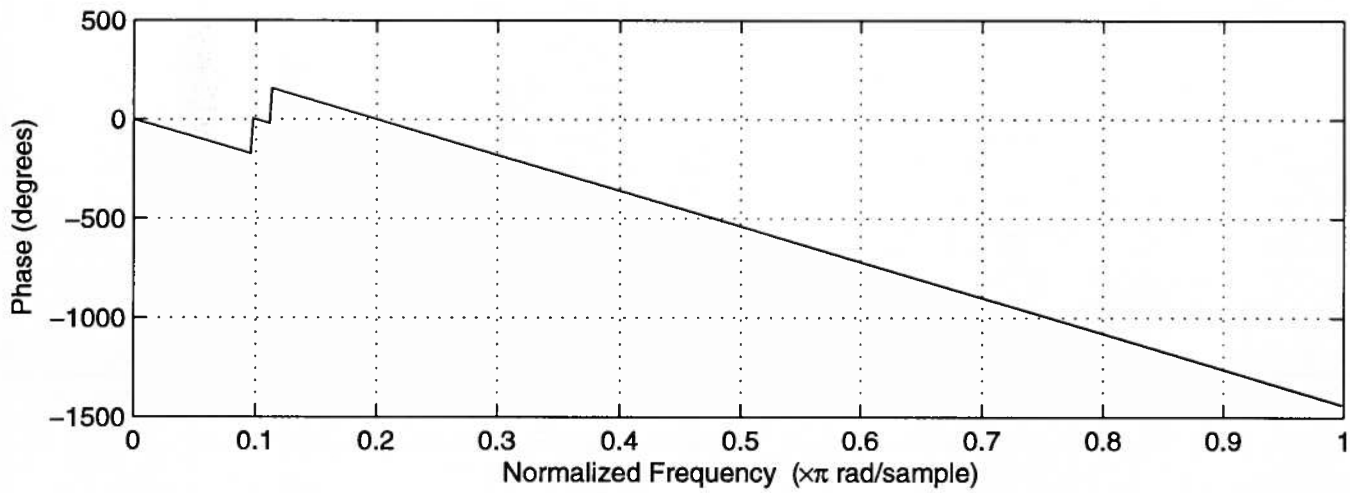
stem (hum1 (1:100))



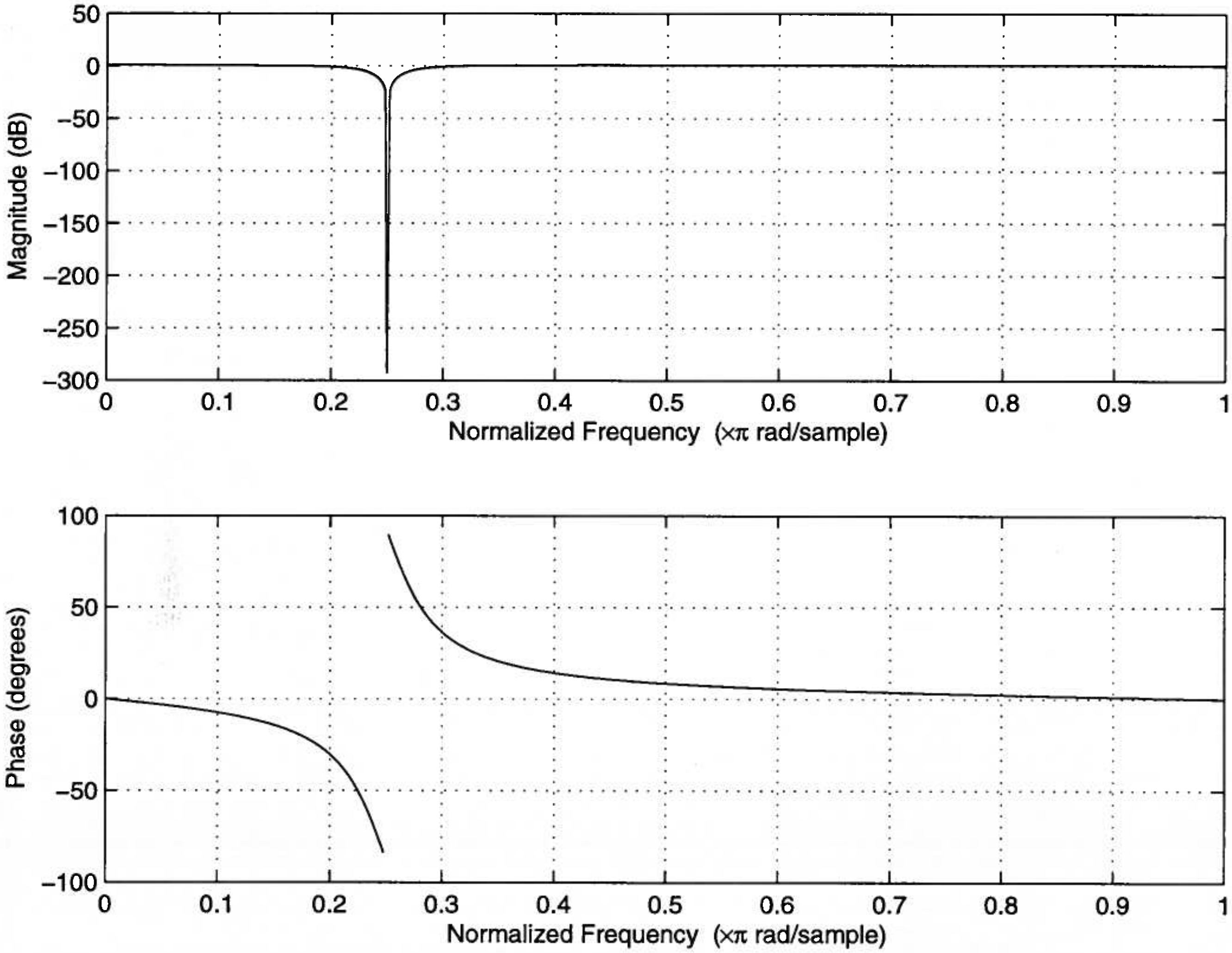
1 7b)

freqz (firnotch)

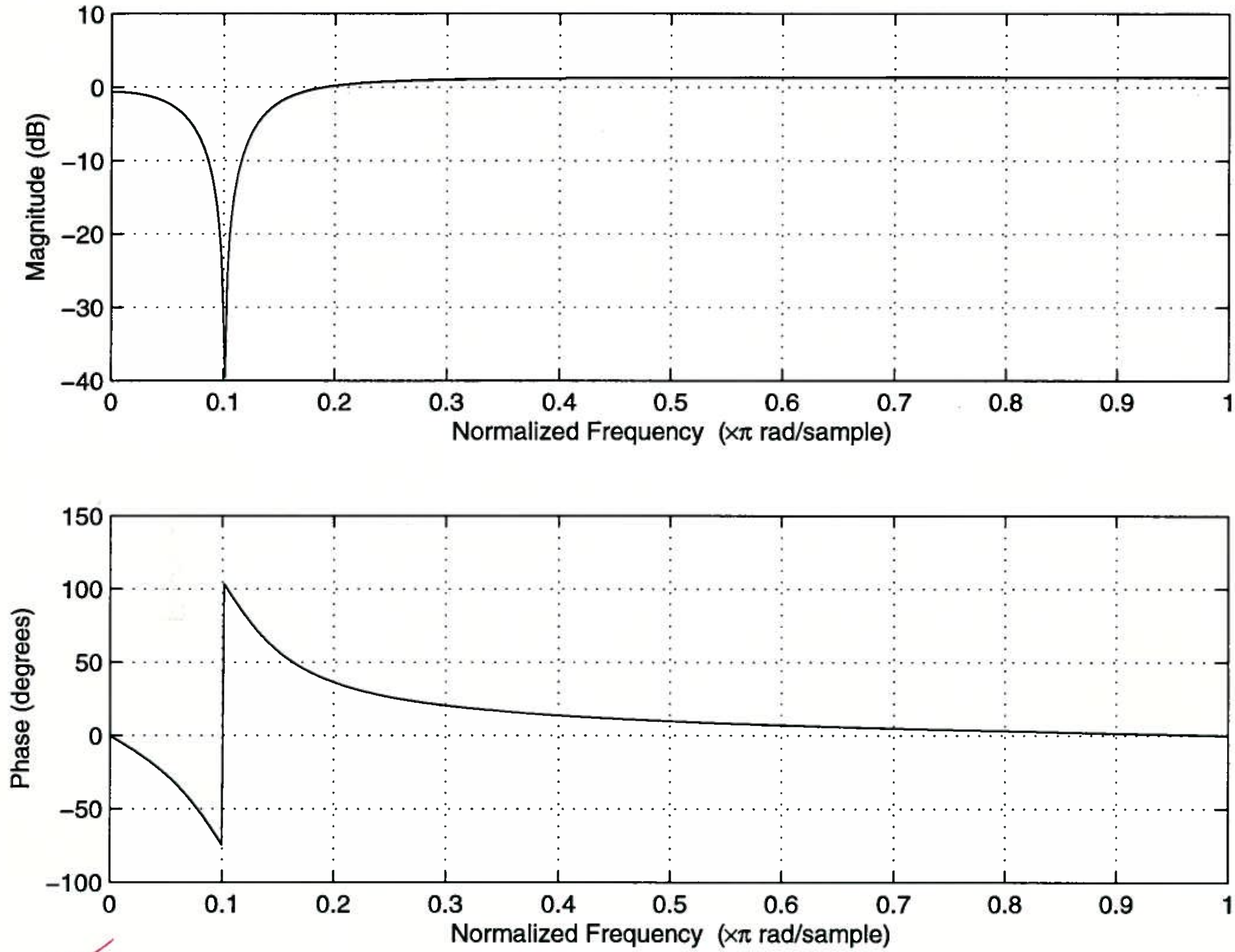




IIR filter w/ $A = 0.9$
$B = \sqrt{2}$



IIR filter w/ A = 0.85
B = 1.9



✓

show the matlab commands for every plot you make.