SigSys Problem Set #6 3/5 Jay Woo

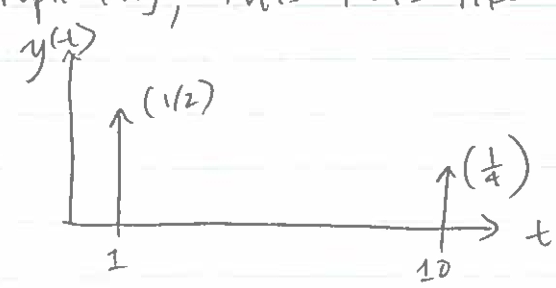
1 The gunshot sound is a very good approximation of an impulse, since it is extremely loud at $t=0$ and quiet at all other times. The impulse response is thus the violin recording convolved with the gunshot impulse, producing a sound as though the violin were being played in the firing range. All frequencies are present in the spectrum of the gunshot audio, so when the gun is fired in the shooting range, the impulse response tells you how each amplitude at each frequency is altered.

2 It is reasonable to call the equation $y(t) = \frac{1}{2}x(t-1) + \frac{1}{4}x(t-10)$ an echo chamber because what it does is it adds two versions of the input signal, with a time difference of 9 units between each wave. You will hear one louder version of the signal at first, and then some time later, the same signal is played but it is much quieter. This simulates the echo.

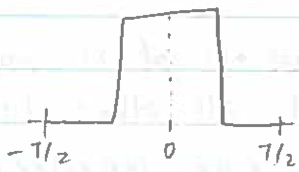
The expression of the impulse response is:

$$y(t) = \frac{1}{2}\delta(t-1) + \frac{1}{4}\delta(t-10)$$

Graphically, this looks like the following:



3 a) The signal can be represented as a piecewise function.



$$\longrightarrow y(t) = \begin{cases} 1 & \text{if } -T/4 \le t \le T/4 \\ 0 & \text{if } \begin{array}{l} t < -T/4 \\ t > T/4 \end{array} \end{cases}$$

The coefficients can be calculated:

$$c_k = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\, e^{-j\frac{2\pi}{T}kt}\, dt$$ you can ignore where $y(t) = 0$

$$= \frac{1}{T}\int_{-T/4}^{T/4} x(t)\, e^{-j\frac{2\pi}{T}kt}\, dt$$

$$= \frac{1}{T}\left(\frac{-T}{2\pi j k}\right) e^{-j\frac{2\pi}{T}kt}\Big|_{-T/4}^{T/4}$$

$$= -\frac{1}{2\pi j k}\left(e^{\frac{-j\pi k}{2}} - e^{\frac{j\pi k}{2}}\right)$$

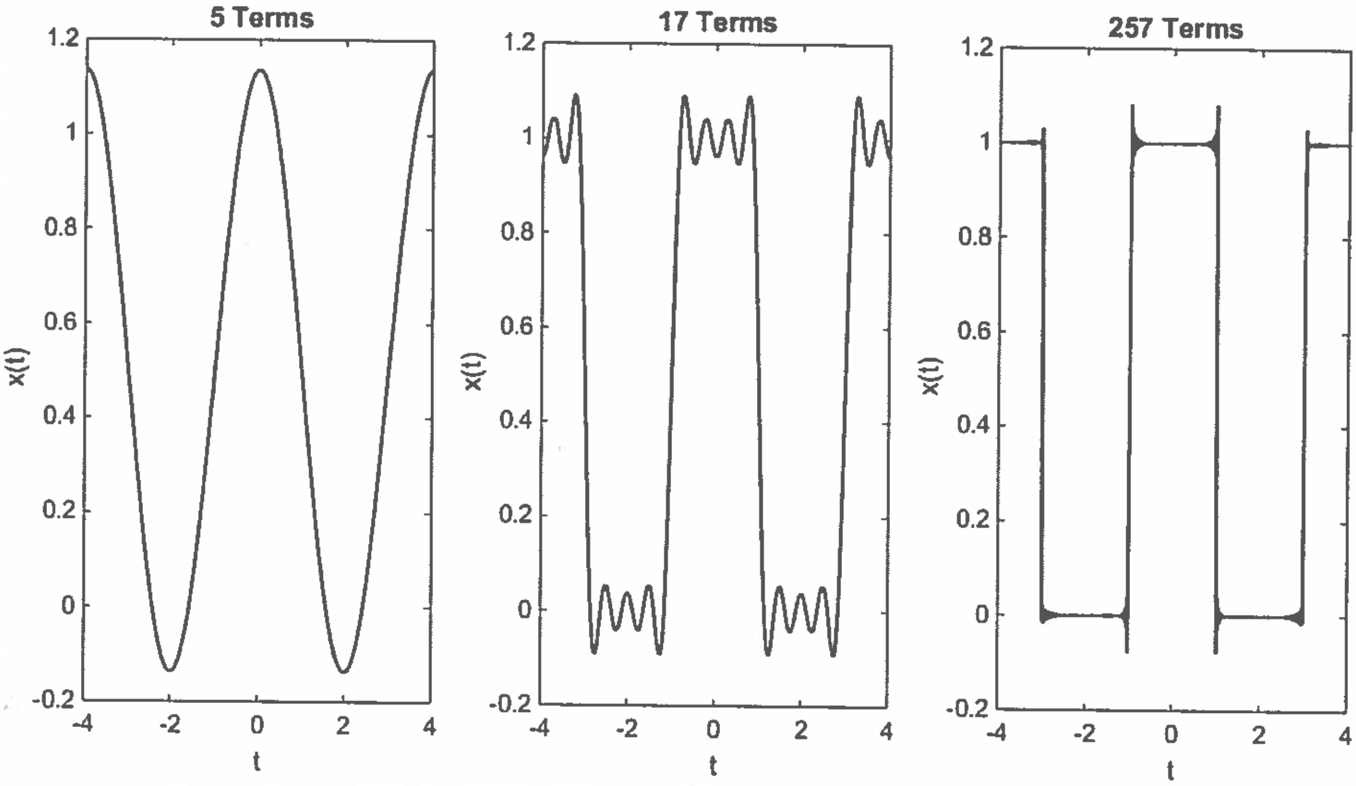
$$= \frac{1}{\pi k}\left(\frac{1}{2j}e^{j\pi k/2} - \frac{1}{2j}e^{-j\pi k/2}\right)$$

$$= \frac{\sin(\pi k/2)}{\pi k} = \frac{\sin(\pi k/2)/2}{\pi k/2} = \mathrm{sinc}\left(\frac{k}{2}\right)/2$$

The full Fourier series:

$$\tilde{x}_k(t) = \sum_{k=-K}^{K} c_k e^{j\frac{2\pi}{T}kt} = \sum_{k=-K}^{K} \frac{\mathrm{sinc}(\frac{k}{2})}{2}\left(e^{j\frac{2\pi}{T}kt}\right)$$

b)



c) For the Fourier series w/ 257 terms, the corners appear to be extremely jagged, and despite the huge number of terms, it is difficult to characterize these huge points of discontinuity. According to equation (10), ...

$$\lim_{K \to \infty} \int_{-T/2}^{T/2} |x(t) - \tilde{x}_K(t)|^2 \, dt = 0$$

K needs to be much higher in order to get the corners "just right" and to reduce the amount of error

The error signal has high energy at the discontinuities

4 a) Given the old set of coefficients $C_k = \int_{-T/2}^{T/2} x(t) e^{-j\frac{2\pi}{T}kt} dt$, the new set of coefficients C_k' is the following:

$$C_k' = \int_{-T/2}^{T/2} x(t-T_1) e^{-j\frac{2\pi}{T}kt} dt \longrightarrow \tau = t - T_1$$

$$= \int_{-T/2}^{T/2} x(\tau) e^{-j\frac{2\pi}{T}k(\tau+T_1)} d\tau$$

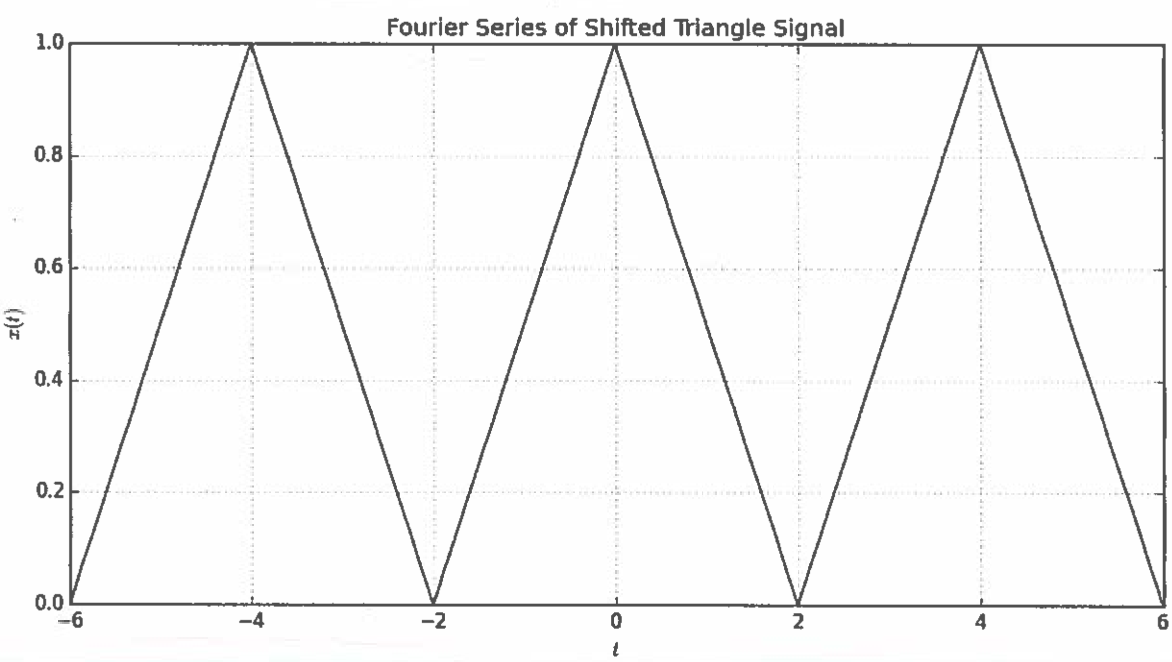
$$= \underbrace{\int_{-T/2}^{T/2} x(\tau) e^{-j\frac{2\pi}{T}k\tau}}\cdot e^{-j\frac{2\pi}{T}kT_1} d\tau$$

$$= C_k e^{-2\pi jk(T_1/T)}$$

The original coefficients get scaled by the expression:

$$\boxed{e^{-2\pi jk(T_1/T)}}$$

b)



Code for #3


```
T = 4;
t = linspace(-4, 4, 500);

% Plots the first 5 terms of the Fourier series
sum = zeros(1, 500);
k = 2;
for i = -k:k
    sum = sum + exp(1j * 2 * pi * i * t / T) * sinc(i /2) / 2;
end

subplot(1,3,1);
plot(t, sum);

% Plots the first 17 terms of the Fourier series
sum = zeros(1, 500);
k = 8;
for i = -k:k
    sum = sum + exp(1j * 2 * pi * i * t / T) * sinc(i /2) / 2;
end

subplot(1,3,2);
plot(t, sum);

% Plots the first 257 terms of the Fourier series
sum = zeros(1, 500);
k = 128;
for i = -k:k
    sum = sum + exp(1j * 2 * pi * i * t / T) * sinc(i /2) / 2;
end

subplot(1,3,3);
plot(t, sum);
```

Code for #4

```python
# Computes the Fourier series of a triangle wave that has been shifted by T1
def fs_triangle_shifted(ts, M=3, T=4.0, T1=2.0):

    # create an array to store the signal
    x = np.zeros(len(ts))
    Coeff = 0

    # if M is even
    if np.mod(M,2) ==0:
        for k in range(-int(M/2), int(M/2)):
            # if n is odd compute the coefficients
            if np.mod(k, 2)==1:
                Coeff = -2/((np.pi)**2*(k**2))
            if np.mod(k,2)==0:
                Coeff = 0
            if n == 0:
                Coeff = 0.5

            Coeff *= np.exp(-1j*2*np.pi*k*(T1/T)) # <------ Shifting factor

            x = x + Coeff*np.exp(1j*2*np.pi/T*k*ts)

    # if M is odd
    if np.mod(M,2) == 1:
        for k in range(-int((M-1)/2), int((M-1)/2)+1):
           # if n is odd compute the coefficients
            if np.mod(k, 2)==1:
                Coeff = -2/((np.pi)**2*(k**2))
            if np.mod(k,2)==0:
                Coeff = 0
            if k == 0:
                Coeff = 0.5
            Coeff *= np.exp(-1j*2*np.pi*k*(T1/T)) # <------ Shifting factor

            x = x + Coeff*np.exp(1j*2*np.pi/T*k*ts)

    return x

ts = np.linspace(-6,6,2048)
x = fs_triangle_shifted(ts, M=127)

mplib.plot(ts, x)
mplib.grid()
mplib.xlabel('$t$')
mplib.ylabel('$x(t)$')
mplib.title('Fourier Series of Shifted Triangle Signal')
```