1. First we need to characterize the room that the gunshot was fired in. If we fire a gun, we provide an impulse. The system, or in this case the room, outputs an impulse response. This is what we hear instantly after firing.

Impulse $\rightarrow$ $\boxed{\text{System/Room}}$ $\longrightarrow$ impulse response

Then, we can take the fourier transform of the impulse response in order to find the transfer function, or the equivalent of the impulse response in the frequency domain. Then we can elementwise multiply the transfer function of the gunshot with the violin wave. Since elementwise multiplication ~~is the~~ in the frequency domain is the same as convolution in the time domain, instead of taking the fourier transform of the impulse response, we can just convolve the impulse response with the violin wave. This will tell us what the violin will sound like in the same room the gun was shot in. Once we have the impulse response, we know what almost any wave will sound like in that room.

2. This is called an echo channel because it will sound like an echo. At $t=1$, the sound will have an amplitude of $1/2$ and later at $t=10$, the sound will have an amplitude of $1/4$. At different times apart, we will hear the same noise but with linearly decreasing

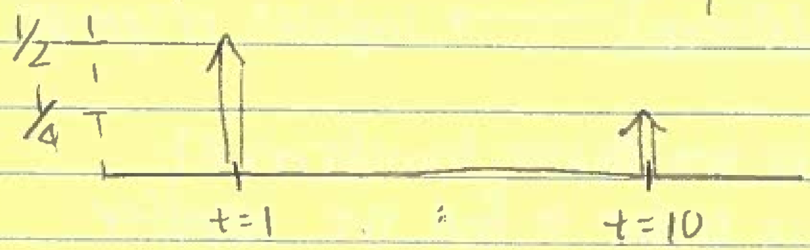
amplitudes, hence, sounding like an echo.

we can say that $y(t) = h(t)$ when $x(t) = \delta(t)$, so

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

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

The impulse response can be plotted:



3a. $x(t) = \frac{4}{T}|t| \longrightarrow x(t) = \begin{cases} 1 & \text{when } -\frac{T}{4} \leq t \leq \frac{T}{4} \\ 0 & \text{otherwise} \end{cases}$

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

$$x_k(t) = \sum_{k=-K}^{K} C_K e^{j\frac{2\pi}{T}kt}$$

$$\to C_K = \frac{1}{T}\int_{-T/2}^{-T/4} 0\, e^{-j\frac{2\pi}{T}kt}\, dt + \frac{1}{T}\int_{-T/4}^{T/4} 1 \cdot e^{-j\frac{2\pi}{T}kt}\, dt + \int_{T/4}^{T/2} 0 \cdot e^{-j\frac{2\pi}{T}kt}\, dt$$

$$C_K = \frac{1}{T}\cdot\int_{-T/4}^{T/4} 1 \cdot e^{-j\frac{2\pi}{T}kt}\, dt = \frac{1}{T}\,\frac{1}{-j\frac{2\pi}{T}k}\cdot e^{-j\frac{2\pi}{T}kt}\Big|_{-T/4}^{T/4}$$

$$C_K = \frac{1}{\cancel{T}}\left[\frac{1}{-j\frac{2\pi}{\cancel{T}}k} e^{-j\frac{\cancel{2}\pi}{\cancel{T}}k(\cancel{T}/\cancel{4})} - \frac{1}{-j\frac{2\pi}{\cancel{T}}k} e^{-j\frac{\cancel{2}\pi}{\cancel{T}}k(-\cancel{T}/\cancel{4})}\right]$$

$$C_K = \frac{-1}{j2\pi k} e^{-j\frac{\pi k}{2}} + \frac{1}{j2\pi k} e^{+j\frac{\pi k}{2}} = \frac{1}{j2\pi k} e^{j\frac{\pi k}{2}} - \frac{1}{j2\pi k} e^{-j\frac{\pi k}{2}}$$

$$\sin(\theta) = \frac{1}{2j} e^{j\theta} - \frac{1}{2j} e^{-j\theta}$$

$$C_K = \frac{1}{\pi k}\left[\sin\left(\frac{\pi k}{2}\right)\right] \longrightarrow C_K = \text{Sinc}\left(K/2\right)\cdot \frac{1}{2}$$

$$x_k(t) = \sum_{k=-K}^{K} \frac{\sin\left(\frac{\pi k}{2}\right)}{\pi k}\cdot e^{j\frac{2\pi}{T}kt} = \sum_{k=-K}^{K} \frac{\sin(K/2)}{2}\cdot e^{j\frac{2\pi}{T}kt}$$

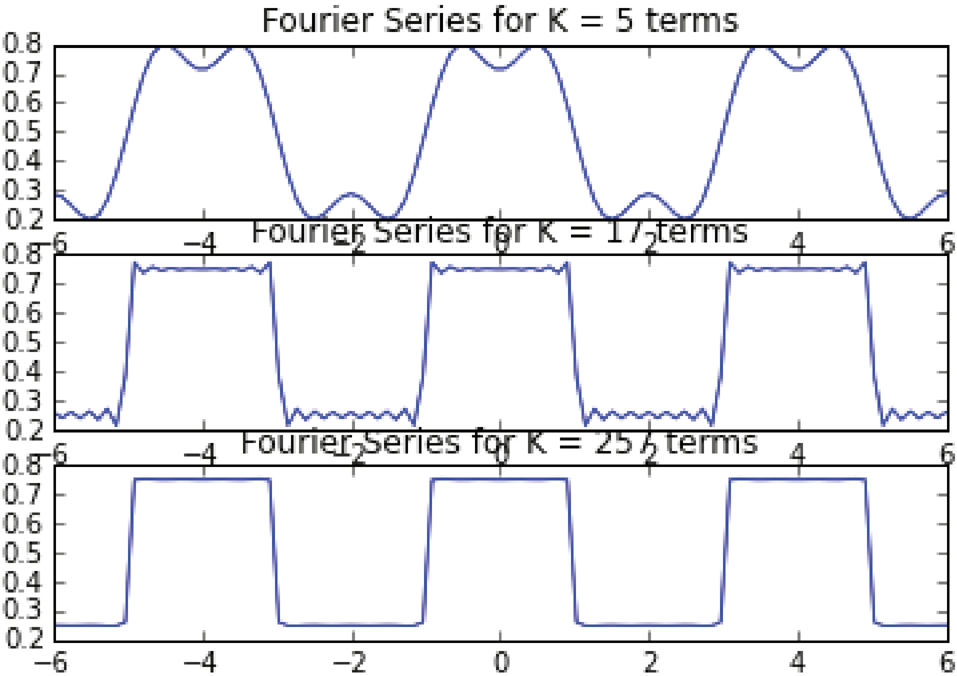
3b. Plotted

3c. At these high points of discontinuity, the jump goes from 0 to 1 and then 1 to 0. This means there is a high energy happened in a short amount of time. the discontinuties ~~happen for~~ for high frequencies. We can fix this error, by increasing the number of terms. This would smooth out higher frequencies by reaching higher frequencies. Reaching higher frequencies would make our approximation

~~More accurate because we~~ more accurate because it would minimize the difference between the actual and the approximation. Equation 10 states that:

$$\int_{-T/2}^{T/2} \left| x(t) - \tilde{x}_k(t) \right|^2 dt$$

If we take the limit as the number of terms approaches infinity, the equation above would decrease and reach a very small number. This means the error would decrease with more frequency terms.



Fourier Series for K = 5 terms
Fourier Series for K = 17 terms
Fourier Series for K = 257 terms

4a.) $x(t) \leftarrow$ periodic, period $= T$, coefficients C_k

$y(t) = x(t - T_1) \qquad |T_1| < T$

$$x_k(t) = \sum_{k=-K}^{K} C_k e^{j\frac{2\pi}{T}kt} \qquad x_k(t - T_1) = \sum_{k=-K}^{K} C_k e^{j\frac{2\pi}{T}k(t-T_1)}$$

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

$T_1 < t$

$y(t) = x(t - T_1)$

$$C_{ky} = \frac{1}{T}\int_{-T/2}^{T/2} x(t - T_1) e^{-j\frac{2\pi}{T}kt}\, dt \qquad j = t - T_1 \quad t = j + T_1$$

$$C_{ky} = \frac{1}{T}\int_{-T/2}^{-T_1} x(t - T_1) e^{-j\frac{2\pi}{T}kt}\, dt + \int_{-T_1}^{T_1} x(t - T_1) e^{-j\frac{2\pi}{T}kt} +$$

$\tau = t - T_1 \rightarrow t = \tau + T_1$

$$C_{ky} = \int_{-T_1}^{T_1} x(\tau) e^{-j\frac{2\pi}{T}k(\tau + T_1)} = \int_{-T_1}^{T_1} x(\tau) e^{-j\frac{2\pi}{T}k\tau} \cdot e^{-j\frac{2\pi}{T}kT_1}\, d\tau$$

$$C_{ky} = \int_{-T_1}^{T_1} x(\tau) \cdot e^{-j\frac{2\pi}{T}k\tau}\, d\tau \cdot e^{-j\frac{2\pi}{T}kT_1} = C_{kx} \cdot e^{-j\frac{2\pi}{T}kT_1}$$

$$C_{ky} = C_{kx} \cdot e^{-j\frac{2\pi}{T}kT_1} \qquad y_s \cdot C_{kx}\, e^{-j\frac{2\pi}{T}T_1}$$

4b.) Plotted on another page

Since $C_{ky} = C_{kx} \cdot e^{-j\frac{2\pi}{T}kT_1}$ and $C_{ky} = \begin{cases} \frac{-2}{\pi^2 k^2} & \text{if } k \text{ is odd} \\ +\frac{1}{2} & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}$

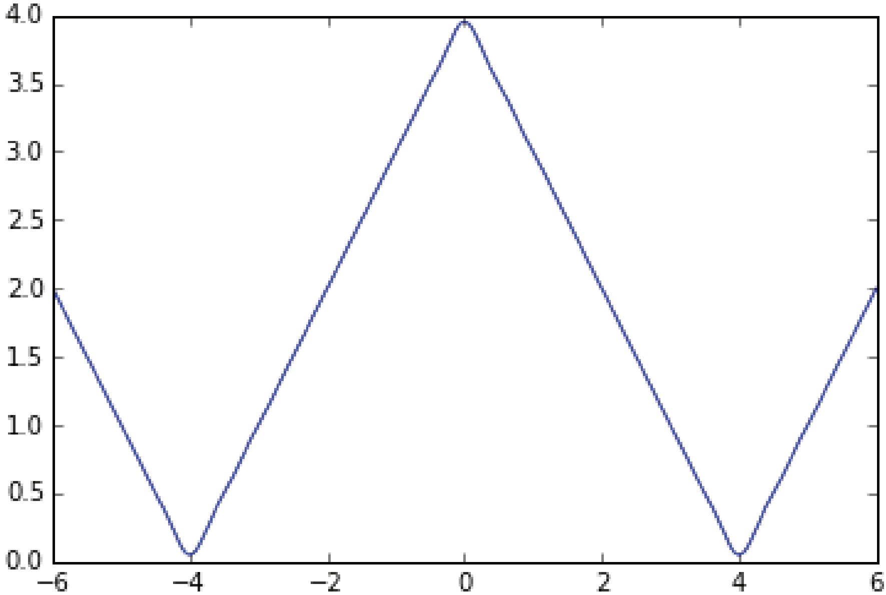
we can say that the fourier series of the triangle wave is:

$$C_{ky} = \begin{cases} \frac{-2}{\pi^2 k^2} \cdot e^{-j\frac{2\pi}{T}kT_1} & \text{if } k \text{ is odd} \\ \frac{1}{2} \cdot e^{-j\frac{2\pi}{T}kT_1} & \text{if } k = 0 \\ 0 \cdot e^{-j\frac{2\pi}{T}kT_1} & \text{otherwise} \end{cases}$$

In [177]:

```
def fs_triangle(ts, M=31, T=4):
    # computes a fourier series representation of a triangle wave
    # with M terms in the Fourier series approximation
    # if M is odd, terms -(M-1)/2 -> (M-1)/2 are used
    # if M is even terms -M/2 -> M/2-1 are used

    # create an array to store the signal
    x = np.zeros(len(ts))
    T_old = T
    T_new = T/2

    # 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))
                Coeff= (T_old)*Coeff
            if np.mod(k,2)==0:
                Coeff = 0
                Coeff= (T_old)*Coeff
            if k == 0:
                Coeff = 0.5
                Coeff= (T_old)*Coeff
            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))
                Coeff= (T_old)*Coeff
            if np.mod(k,2)==0:
                Coeff = 0
                Coeff= (T_old)*Coeff
            if k == 0:
                Coeff = 0.5
                Coeff= (T_old)*Coeff

            x = x + Coeff*np.exp(1j*np.pi/T*k*ts)*np.exp(-1j*(2*np.pi/T_old)*k*T_new)
#     return x
    return x
```

Below is the triangle wave from Figure 2, unaltered.



Below is the triangle wave shifted using the fourier coefficients found in problem 4a.