① $\delta(n) \longrightarrow \boxed{TF1} \longrightarrow h(n)$

Gunshot   Gunshot impulse response

the Violin recording was centered around the impulse response and scaled to its amplitude

$x(n) \longrightarrow \boxed{TF2} \longrightarrow y(n)$

Violin

transfer function 2

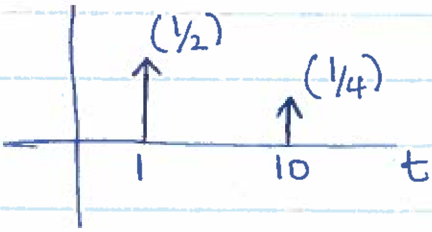
$$y(n) = \underbrace{x * h(n)}_{\text{convolution of gunshot impulse response and violin recording}}$$

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

Reasonable to call it an echo channel:

→ decreasing amplitude $(\frac{1}{2} \rightarrow \frac{1}{4})$

→ delayed time (1s, 10s)



③ $\sin\theta = \frac{1}{2j}e^{j\theta} - \frac{1}{2j}e^{-j\theta}$

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

$x(t) = 1$

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

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

$$C_k = \frac{1}{\pi k}\left[\frac{-1}{2j}e^{-j\frac{\pi}{2}k} + \frac{1}{2j}e^{j\frac{\pi}{2}k}\right.$$

$$C_k = \frac{1}{\pi k}\sin\left(\frac{\pi}{2}k\right)$$

if k is even, $C_k = 0$

if k is odd, $C_k = \frac{1}{\pi k}\sin$

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

$$= \sum_{k=-K}^{K} \frac{1}{\pi k}\left(\sin\left(\frac{\pi}{2}k\right)\right) e^{j\frac{2\pi}{T}kt}$$

$$x(t) = \sum_{k=-\infty}^{\infty} C_k e^{j\frac{2\pi}{T}kt}$$

when $k \neq m$,

$$\frac{1}{T}\int_{-T/4}^{T/4} x(t) e^{-j\frac{2\pi}{T}mt}\,dt = \sum_{k=-\infty}^{\infty} \frac{C_k}{T}\int_{-T/4}^{T/4} e^{j\frac{2\pi}{T}(k-m)t}\,dt$$

$$\int_{-T/4}^{T/4} e^{j\frac{2\pi}{T}(k-m)t}\,dt = \frac{1}{j\frac{2\pi}{T}(k-m)}\left[e^{j\frac{2\pi}{T}(k-m)t}\right]_{-T/4}^{T/4}$$

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

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

$$= \frac{1}{\frac{\pi}{T}(k-m)} \sin\left(\frac{\pi}{2}(k-m)\right) = 0$$

when $k = m$,

$$\int_{-T/4}^{T/4} e^{j\frac{2\pi}{T}(k-m)t}\,dt = \int_{-T/4}^{T/4} 1\,dt = \left[\frac{T}{4}+\frac{T}{4}\right] = \frac{T}{2}$$

$$\frac{1}{T}\int_{-T/4}^{T/4} e^{j\frac{2\pi}{T}lt}\,dt = \begin{cases} \frac{1}{2} & \text{if } l = 0 \\ 0 & \text{otherwise} \end{cases}$$

k=m is the only term in the summation because all other numbers/terms cancel each other out.

$$\frac{1}{T}\int_{-T/4}^{T/4} x(t)e^{-j\frac{2\pi}{T}mt}dt = \frac{1}{T}\int_{-T/4}^{T/4}\sum_{k=-\infty}^{\infty} C_k e^{j\frac{2\pi}{T}(k-m)t}dt$$

$$= \frac{C_m}{T}\int_{-T/4}^{T/4} e^{j\frac{2\pi}{T}(m-m)t}dt$$

$$= \frac{C_M}{2}$$

In the graphs, where the square wave goes from 0 to 1, there are spikes as the signal ~~overshoots~~ oscillates around the point (0 and 1).

~~[illegible]~~ ~~Since~~ There is a discontinuity where the points jump from 0 to 1 a[nd] 1 to 0. ~~the~~ Equation (10) tries to approximate verify the ~~diffe~~[rence] of function x(t). However, at discontinuous points, even w[hen] the frequencies increase (approach ∞), the ~~[illegible]~~ limit doe[s] not converge at 0 or 1, it converges slightly higher/low[er] due to the e[...]

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

Assume $x(t) = 1$

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

$$= \frac{1}{\cancel{T}}\left[\frac{-\cancel{T}}{j2\pi k}e^{-j\frac{2\pi kt}{T}}\right]_{-T/2-T_1}^{T/2-T_1}$$

$$= \frac{1}{j2\pi k}\left[e^{-j\frac{2\pi k}{T}(T/2-T_1)} + e^{-j\frac{2\pi k}{T}(-T/2-T_1)}\right]$$

$$= \frac{e^{j\frac{2\pi k T_1}{T}}}{\pi k}\sin(\pi k)$$

sinc function multiplied by a constant

$$a = e^{j\frac{2\pi k T_1}{T}}$$

In the triangle wave,

$T_1 = \frac{T}{2}$

so the constant is $e^{j\pi k} = a$

when k is even, constant $a = 1$

when k is odd, constant $a = -1$

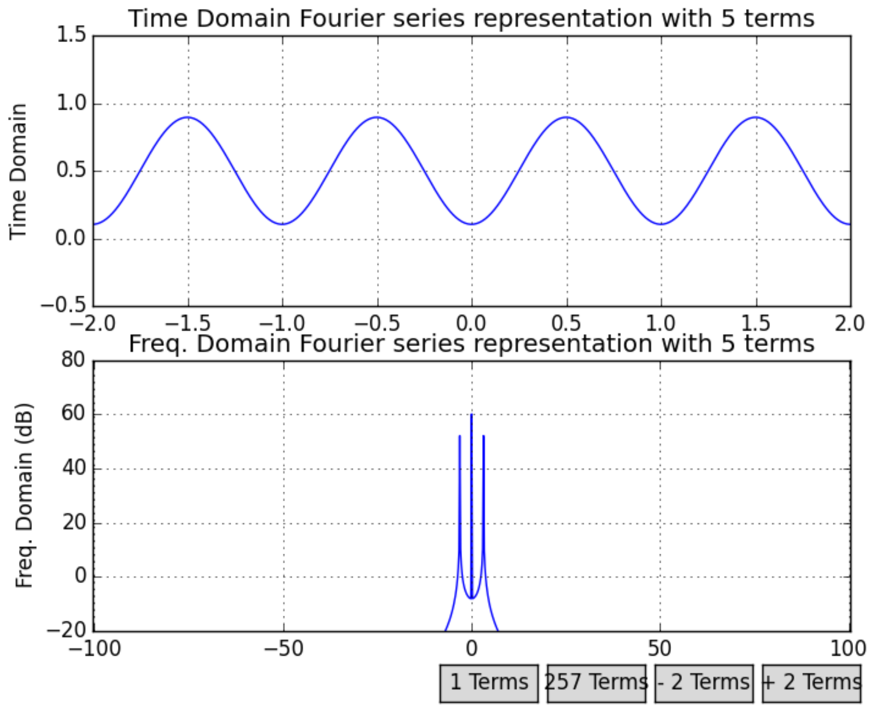
so...

C_k for the shifted triangle wave is...

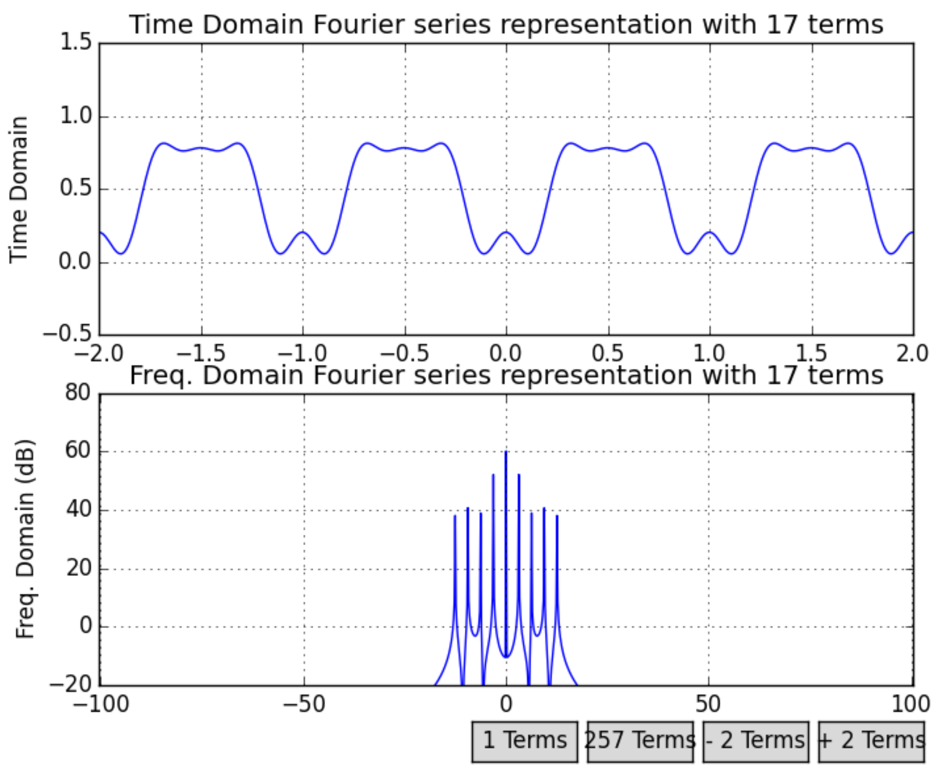
$$\boxed{\begin{array}{ll} \frac{2}{\pi^2 k^2} & \text{for odd } k\text{'s} \\ 0.5 & \text{for even } k\text{'s} \\ 0 & \text{otherwise} \end{array}}$$

Square Wave

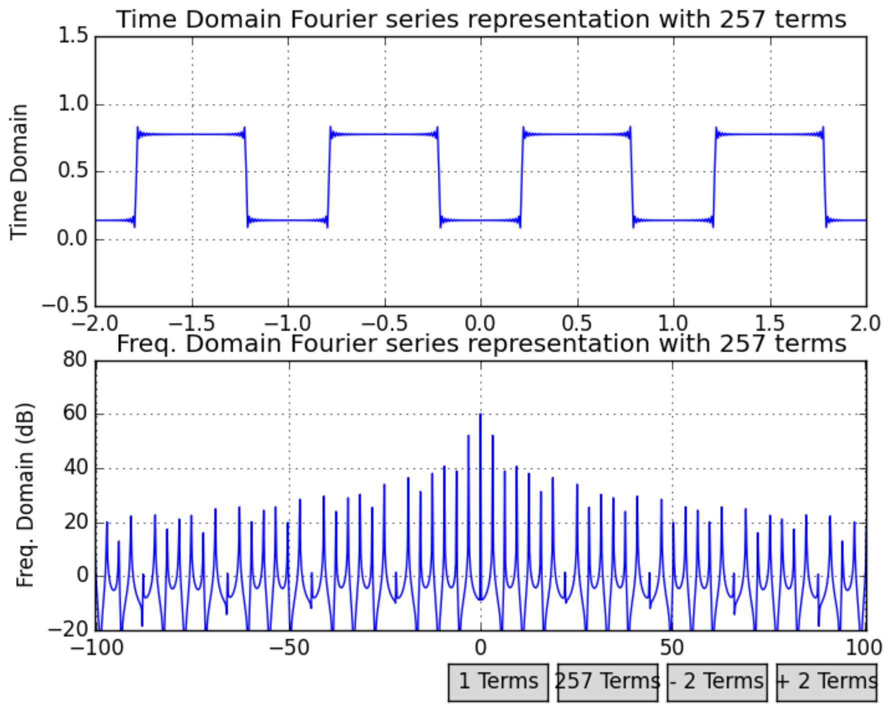
5 Terms



17 Terms



257 Terms



Triangle Wave Code

```
def fs_triangle(ts, M=3, 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))

    # 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 = np.exp(np.pi*k*1j)*-2/((np.pi)**2*(k**2))
            if np.mod(k, 2)==0:
                Coeff = 0
            if n == 0:
                Coeff = np.exp(np.pi*k*1j)*0.5
            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 = np.exp(np.pi*k*1j)*-2/((np.pi)**2*(k**2))
       if np.mod(k, 2)==0:
           Coeff = 0
       if k == 0:
           Coeff = np.exp(np.pi*k*1j)*0.5
       x = x + Coeff*np.exp(1j*2*np.pi/T*k*ts)

return x
```