Date …………………… Experiment No………

PULSE SHAPING AND MATCHED FILTERS

Aim:

To

1. Generate a string of message bits.
2. Use root raised cosine pulse p(t) as the shaping pulse, and generate the corresponding baseband signal with a fixed bit duration Tb. You may use roll-off factor as $\alpha = 0.4$.
3. Simulate transmission of baseband signal via an AWGN channel
4. Apply matched filter with frequency response $P_r(f) = P^*(f)$ to the received signal.
5. Sample the signal at mT_b and compare it against the message sequence.

Theory:

Inter symbol interference (ISI) is a form of distortion of a signal in which the symbol interferes with subsequent symbol.The spreading of the pulse beyond its allotted time interval causes it to interfere with neighboring pulses.ISI degrade the bit and symbol error rate performance in the presence of noise. The causes of ISI are multipath propagation and dispersion of channels. The baseband transmission system is as shown in figure 1.

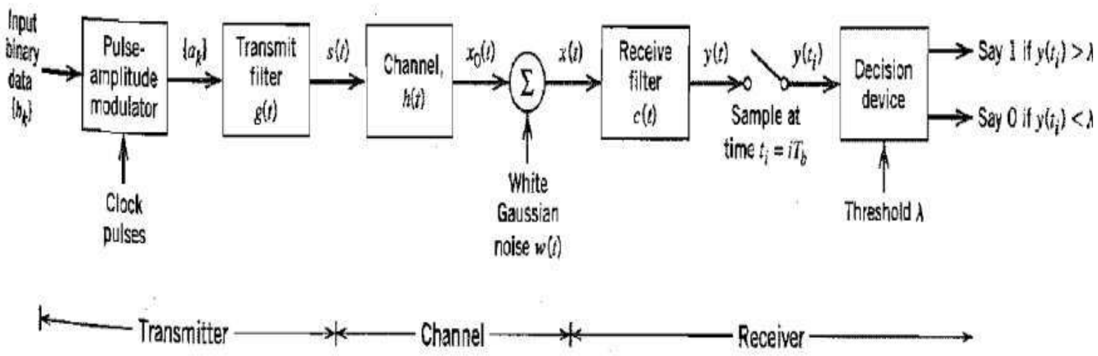


FIGURE 4.7 Baseband binary data transmission system.

Here,

$$ak = \{+1, if\ the\ symbol\ b_k\ is\ 1 - 1, if\ the\ symbol\ b_k\ is\ 0$$

$$s(t) = \sum_k a_k g(t - kT_b)$$

The received filter output is written as

$$y(t) = \mu \sum_k a_k p(t - kT_b) + n(t).$$

Where μ- scaling factor and pulse p(t) is to be defined.

At i^{th} instant

$$y(t_i) = \mu \sum_{k=-\alpha} a_k p[(i-k)T_b] + n(t_i)$$

$$y(t_i) = \mu a_i + \sum_{k=-\alpha, k\neq i} a_k p[(i-k)T_b] + n(t_i)$$

The second term is due to ISI.

To avoid ISI Pulse shaping filters are used.Pulse shaping filter must be chosen carefully not to introduce inter symbol interference. The commonly used pulse shaping filters are –

(i) Rectangular pulse shape: This pulse shape has poor spectral properties with high sidelobes.

(ii) Sinc pulse shape: Theoretically, the sinc filter has ideal spectral properties, as the Fourier transform of a sinc function is an ideal lowpass spectrum. However, a sinc pulse is non-causal, hence not realizable.

(iii) Raised-cosine pulse: This is a pulse widely used in practice. The pulse shape and the excess bandwidth can be controlled by changing the roll-off factor (0≤α≤1, where 0 means no excess bandwidth, and 1 means maximum excess bandwidth)

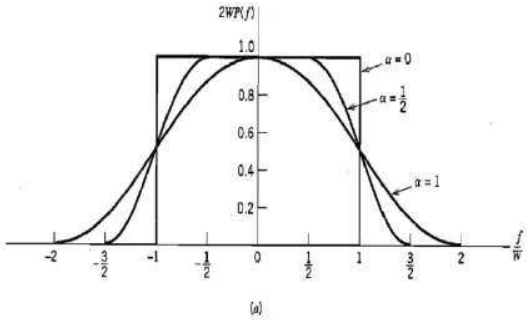


Figure 2. Raised cosine spectrum- frequency spectrum for different α values

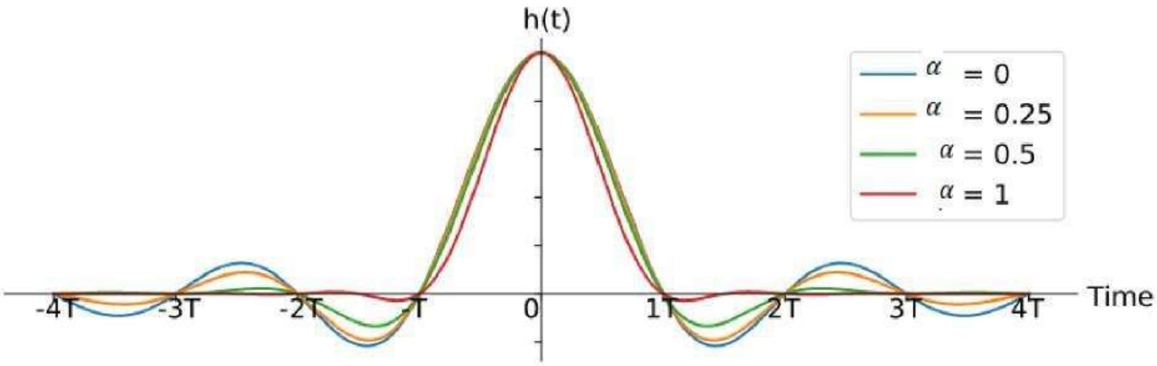


Figure 3. Impulse response of Raised cosine filter with variousα values

$$h(t) = \{\frac{1}{4T} Sinc\left(\frac{1}{2\alpha}\right), \qquad t = \pm\frac{T}{2\alpha}\ \frac{1}{T} sinc\left(\frac{t}{T}\right)\frac{Cos\left(\frac{\pi\alpha t}{T}\right)}{1-(\frac{2\alpha t}{T})^2}, otherwise$$

(iv) Root raised cosine pulse: It has a transfer function equal to square root of raised cosine filter. This filter satisfies Nyquist criteria. These filters are real valued and symmetric. It has its own matched filter.

(v) Gaussian filter: The impulse response of this filter is a Gaussian function. Gaussian pulses have good spectral properties.

Matched filter:

A matched filter is a filter to provide maximum signal to noise ratio at the output.The Characteristic of the matched filter at the receiver should be complex conjugate of the one at the transmitter in order to fulfill Nyquist criteria. If an RRC filter used at the transmitter, the same filter can be used as it is in the receiver since RRC filter is its own matched filter. The impulse response of matched filter is h(t)= s(t-τ). Where s(t) is the input.

Algorithm :

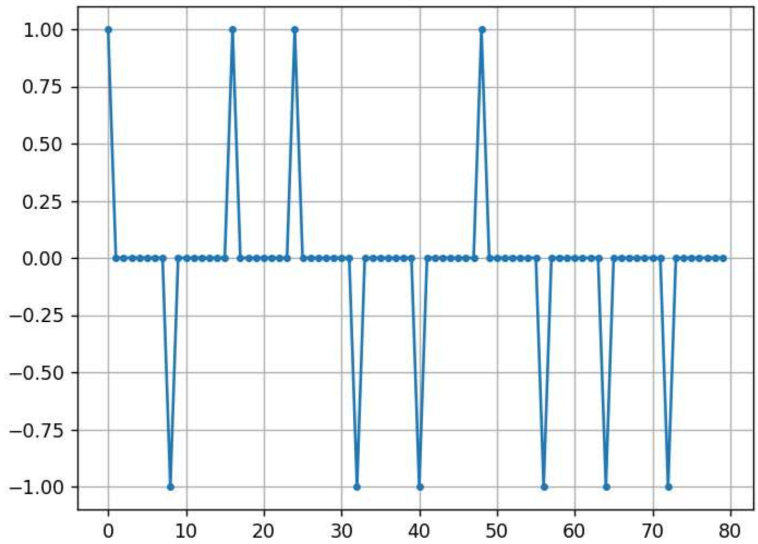
1. Specify the no.of symbols transmitted
2. Specify the no.of samples of transmitted signal.
3. Generate random binary data and convert it to NRZ format.
4. Oversample each bit by adding 8 samples.
5. Assume the number of taps, roll off rate (alpha) and sample period.
6. Generate a Raised Cosine filter.
7. Perform convolution between raised cosine filter and input signal.
8. Simulate an AWGN channel.
9. Send the convoluted signals through AWGN channel.
10. Generate output after convolution of noise affected signal with matched filter response.
11. Plot the output and compare with transmitted bits.

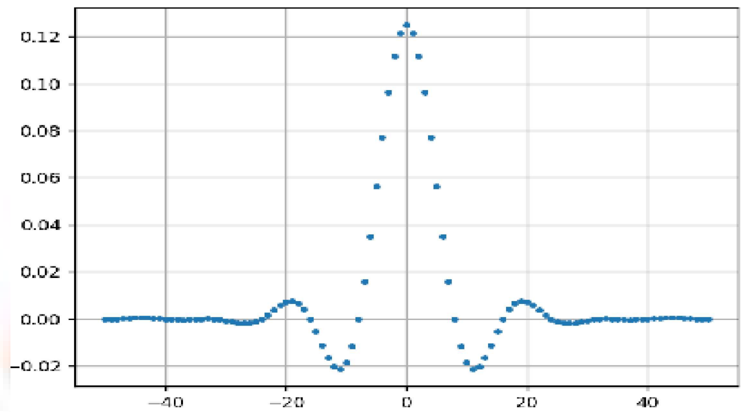
Program :

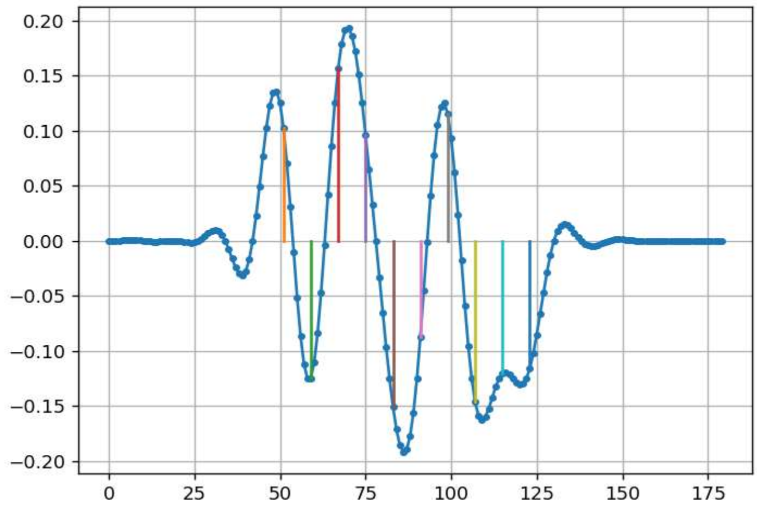
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

num_symbols = 10
sps = 8

bits = np.random.randint(0, 2, num_symbols) # Our data to be transmitted, 1's and 0's

x = np.array([])
for bit in bits:
    pulse = np.zeros(sps)
    pulse[0] = bit*2-1 # set the first value to either a 1 or -1
    x = np.concatenate((x, pulse)) # add the 8 samples to the signal
plt.figure(0)
plt.plot(x, '.-')
plt.grid(True)
plt.show()

# Create our raised-cosine filter
num_taps = 101
beta = 0.35
Ts = sps # Assume sample rate is 1 Hz, so sample period is 1, so *symbol* period is 8
t = np.arange(-50, 51) # remember it's not inclusive of final number
h = 1/Ts*np.sinc(t/Ts) * np.cos(np.pi*beta*t/Ts) / (1 - (2*beta*t/Ts)**2)
plt.figure(1)
plt.plot(t, h, '.')
```

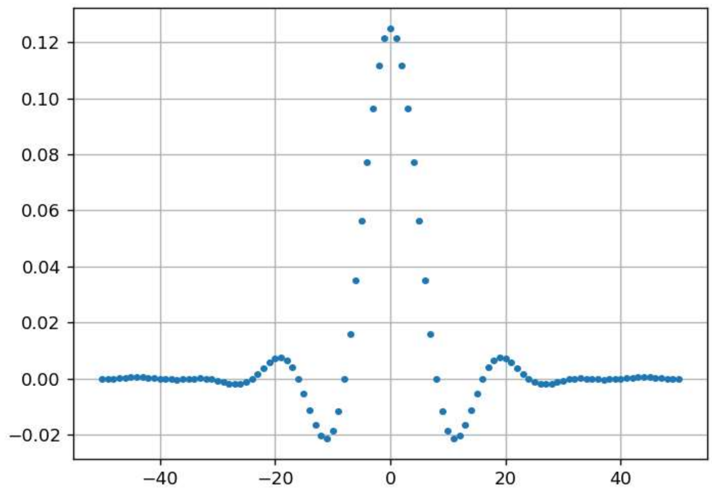
```
plt.grid(True)
plt.show()

x_shaped = np.convolve(x, h)
plt.figure(2)
plt.plot(x_shaped, '.-')
for i in range(num_symbols):
    plt.plot([i*sps+num_taps//2+1,i*sps+num_taps//2+1], [0, x_shaped[i*sps+num_taps//2+1]])
plt.grid(True)
plt.show()
```

Result :


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60
70
80


0.12
0.10
0.08
0.06
0.04
0.02
0.00
−0.02
−40
−20
0
20
40





RESULT:

The performance of raised cosine pulse shaping and matched filter are simulated