Digital Communication Lab
Analysis of Raised Cosine Function and Bandwidth Calculation

Name: Sachin Kumar
Roll No: 220104026
Section: D
Lab No. : 6

October 19, 2024

Objective

The purpose of this lab is to analyze the raised cosine function in both the time and frequency domains and compute the bandwidth for different roll-off factors (α). Additionally, excess bandwidth is calculated based on different values of α.

Introduction

The raised cosine filter is widely used in digital communication systems for pulse shaping to minimize intersymbol interference (ISI). The roll-off factor α controls the bandwidth of the filter. In this lab, we analyze the effect of α on the raised cosine function and calculate the bandwidth and excess bandwidth for different values of α.

Code Implementation

Below is the Python code used to plot the raised cosine function for different values of α and calculate the bandwidth and excess bandwidth.

```
import numpy as np
```

```
import matplotlib.pyplot as plt

# Parameters
T = 1  # Period (you can change this as needed)
t = np.linspace(-2*T, 2*T, 100)  # Time variable over 4 periods

alpha=0
Xrct1=(np.sinc(t/T)*(np.cos(((np.pi)*alpha*t/T)) / (1-(4*alpha*alpha*t*t
alpha=0.5
Xrct2=(np.sinc(t/T)*(np.cos(((np.pi)*alpha*t/T)) / (1-(4*alpha*alpha*t*t
alpha=1
Xrct3=(np.sinc(t/T)*(np.cos(((np.pi)*alpha*t/T)) / (1-(4*alpha*alpha*t*t

plt.plot(t, Xrct1, label='Xrct1: for alpha = 0', color='b')
plt.plot(t, Xrct2, label='Xrct2: for alpha = 0.5', color='r')
plt.plot(t, Xrct3, label='Xrct3: for alpha = 1', color='g')

# Adding labels and title
plt.title('Graphs of Xrct1, Xrct2, and Xrct3')
plt.xlabel('t')
plt.ylabel('Amplitude')

# Show grid and legend
plt.grid(True)
plt.legend()

# Define the function Xrc(f)
def Xrc(f, T, alpha):
    abs_f = np.abs(f)
    if abs_f < (1 - alpha) / 2 * T:
        return T
    elif (1 - alpha) / 2 * T <= abs_f <= (1 + alpha) / 2 * T:
        return (T / 2) * (1 + np.cos(np.pi * (abs_f - (1 - alpha) / 2 *
    else:
        return 0

# Apply the function over a range of frequencies
f_values = np.linspace(-T, T, 1000)
alpha = 0
Xrc_values1 = np.array([Xrc(f, T, alpha) for f in f_values])
alpha = 0.5
Xrc_values2 = np.array([Xrc(f, T, alpha) for f in f_values])
alpha = 1
```

```
Xrc_values3 = np.array([Xrc(f, T, alpha) for f in f_values])
# Plotting
plt.figure(figsize=(10, 6))
plt.plot(f_values, Xrc_values1, label=r'$Xrc(f)$ for alpha=0$', color='b
plt.plot(f_values, Xrc_values2, label=r'$Xrc(f)$ for  alpha=0.5$', color
plt.plot(f_values, Xrc_values3, label=r'$Xrc(f)$ for alpha=1$', color='g
# Add title, labels, and grid
plt.title(r'Graph of $Xrc(f)$ for $\alpha = 0, 0.5, 1$')
plt.xlabel('Frequency (f)')
plt.ylabel('Xrc(f)')
plt.grid(True)
plt.legend()

plt.show()
```

Generated Plots

The following plots show the raised cosine function in the time and frequency domains for different values of α.

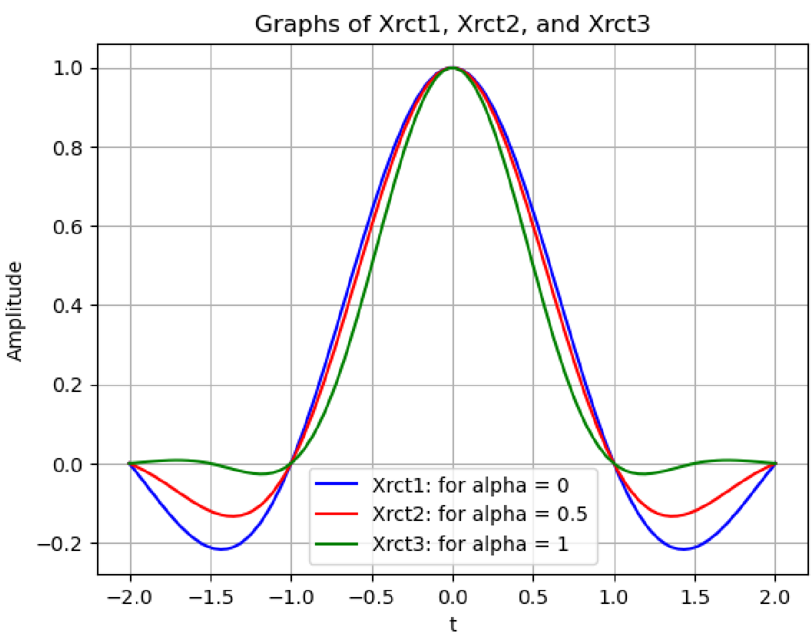

Figure 1: Plot of $Xrct(t)$ for $\alpha = 0, 0.5, 1$

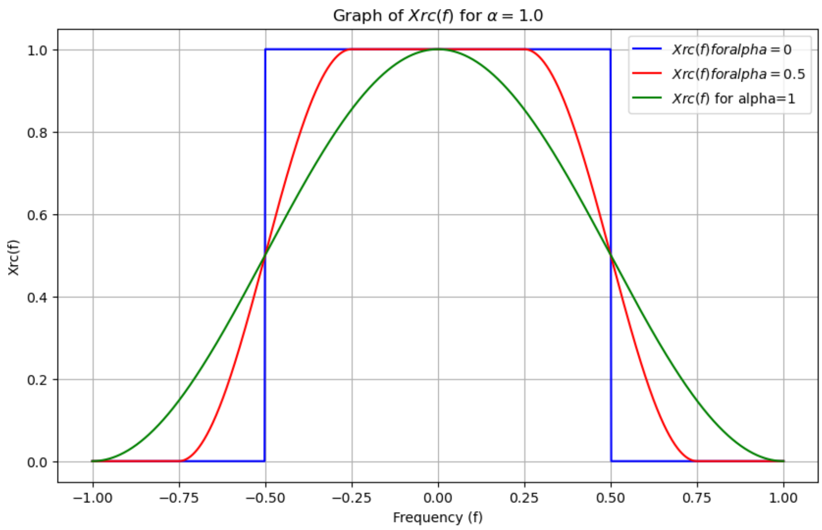

Figure 2: Plot of $Xrc(f)$ for $\alpha = 0, 0.5, 1$

Bandwidth and Excess Bandwidth Calculation

The following Python code calculates the bandwidth and excess bandwidth for different values of α:

Listing 1: Python Code for Bandwidth Calculation

```
# Bandwidth and Excess Bandwidth Calculation
alpha = 0
bandwidth = (1/T) * (1 + alpha)
excessbw = bandwidth - 1/T
excessbw_percentage = (excessbw / (1/T)) * 100
print(f'Bandwidth for alpha = {alpha} is {bandwidth:.1f} Hz')
print(f'Excess bandwidth for alpha = {alpha:.2f} is {excessbw_percentage

alpha = 0.5
bandwidth = (1/T) * (1 + alpha)
excessbw = bandwidth - 1/T
excessbw_percentage = (excessbw / (1/T)) * 100
print(f'Bandwidth for alpha = {alpha} is {bandwidth:.1f} Hz')
print(f'Excess bandwidth for alpha = {alpha:.2f} is {excessbw_percentage

alpha = 1
bandwidth = (1/T) * (1 + alpha)
excessbw = bandwidth - 1/T
excessbw_percentage = (excessbw / (1/T)) * 100
```

```
print(f'Bandwidth for alpha = {alpha} is {bandwidth:.1f} Hz')
print(f'Excess bandwidth for alpha = {alpha:.2f} is {excessbw_percentage
```

Results

The bandwidth and excess bandwidth values for different roll-off factors are shown in the table below:

```
Bandwidth for alpha=0: 1.0
Bandwidth for alpha=0.5: 1.5
Bandwidth for alpha=1: 2.0
```

Figure 3: Plot of $Xrct(t)$ for $\alpha = 0, 0.5, 1$

```
Bandwidth for alpha = 0 is 1.0 Hz
Excess bandwidth for alpha = 0.00 is 0.00%
Bandwidth for alpha = 0.5 is 1.5 Hz
Excess bandwidth for alpha = 0.50 is 50.00%
Bandwidth for alpha = 1 is 2.0 Hz
Excess bandwidth for alpha = 1.00 is 100.00%
```

Figure 4: Plot of $Xrct(t)$ for $\alpha = 0, 0.5, 1$

Conclusion

In this lab, we successfully analyzed the raised cosine function and calculated its bandwidth and excess bandwidth for different values of α. As the roll-off factor α increases, the bandwidth increases, along with the excess bandwidth, which demonstrates the trade-off between bandwidth and spectral efficiency in communication systems.