

Exp-1.3

Natural Sampling

Name: Ahamed Hariz A

Reg no: 212224060009

```
# #Natural sampling
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import butter, lfilter
fs = 1000 # Sampling frequency (samples per second)
T = 1 # Duration in seconds
t = np.arange(0, T, 1/fs) # Time vector
fm = 5 # Frequency of message signal (Hz)
message_signal = np.sin(2 * np.pi * fm * t)
pulse_rate = 50 # pulses per second
pulse_train = np.zeros_like(t)
pulse_width = int(fs / pulse_rate / 2)
for i in range(0, len(t), int(fs / pulse_rate)):
    pulse_train[i:i+pulse_width] = 1
nat_signal = message_signal * pulse_train
sampled_signal = nat_signal[pulse_train == 1]
sample_times = t[pulse_train == 1]
# # Interpolation - Zero-Order Hold (just for visualization)
reconstructed_signal = np.zeros_like(t)
for i, time in enumerate(sample_times):
    index = np.argmin(np.abs(t - time))
    reconstructed_signal[index:index+pulse_width] = sampled_signal[i]
def lowpass_filter(signal, cutoff, fs, order=5):
    nyquist = 0.5 * fs
    normal_cutoff = cutoff / nyquist
    b, a = butter(order, normal_cutoff, btype='low', analog=False)
    return lfilter(b, a, signal)
reconstructed_signal = lowpass_filter(reconstructed_signal, 10, fs)
plt.figure(figsize=(14, 10))
plt.subplot(4, 1, 1)
plt.plot(t, message_signal, label='Original Message Signal')
plt.title("NAME: AHAMED HARIZ A\nREG NO: 212224060009\n\n")
plt.legend()
plt.grid(True)
plt.subplot(4, 1, 2)
plt.plot(t, pulse_train, label='Pulse Train')
plt.legend()
plt.grid(True)
plt.subplot(4, 1, 3)
plt.plot(t, nat_signal, label='Natural Sampling')
plt.legend()
```

```
plt.legend()
plt.grid(True)
plt.subplot(4, 1, 4)
plt.plot(t, reconstructed_signal, label='Reconstructed Message Signal',
color='green')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

NAME: AHAMED HARIZ A
REG NO: 212224060009

