



Escola Politécnica
Disciplina: Redes 2
Professor: Felipe Viel

Relatório trabalho referente ao FSK

Acadêmicos:
Bruno Pereira Freitas
Matheus da Costa Raimundo

ITAJAÍ 2023

Introdução

Neste relatório será contemplado as experiências obtidas no desenvolvimento do processo de modularização e demodularização com base em FSK, a princípio o prof. Felipe Viel disponibilizou um código base a ser utilizado teria partes faltantes que deveriam ser desenvolvidos pelos alunos regentes deste trabalho. O resultado teria que ser composto de 2 gráficos demonstrando os resultados do espectrograma da frequência a ser passada, também a mensagem modularizada em binário e depois a demodularizada e passada para ascii novamente tornando legível novamente.

Modularização em FSK

FSK (Frequency-Shift Keying) é uma técnica de modulação digital em que a informação é transmitida através da variação de frequência da portadora. Na FSK binária, é usada uma frequência para representar o bit 0 e outra frequência para representar o bit 1. Na demodulação, é possível identificar qual das frequências foi recebida e, assim, recuperar a informação transmitida. A FSK é amplamente utilizada em sistemas de comunicação digital como telefonia, rádio e satélite.

Código

O código base foi cedido pelo professor, nessa imagem pode-se reparar nas bibliotecas utilizadas, como foi citado que o uso de qualquer biblioteca referente a FSK resultaria em desqualificação do trabalho optamos por deixar as bibliotecas que já estavam na base.

```
import matplotlib.pyplot as plt
import numpy as np
from math import pi
from scipy.fft import fft, fftfreq
import pandas as pd
import binascii

# criação de seno
def cont_sin(time, sample_rate, frequency):
    time = time
    sample = sample_rate
    f = frequency
    t = np.linspace(time, time+0.1, sample)
    signal = np.sin(2*np.pi*f*t)
    return t, signal
```

Abaixo das bibliotecas tem uma função para a criação do Seno como pode ser visto na imagem a cima.

Nessa parte foi eleito os padrões da frequência, que veio do código base, tanto quanto a mensagem teste 'hello comp'

```
# -----MODULAÇÃO-----

# padrões de frequencia
Fs = 10000
T = 0
fc1 = 1300 # 1
fc2 = 1700 # 0

# construindo mensagem
ascii_message = 'hello comp'
bin_message = ''.join(format(ord(i), '08b') for i in ascii_message)
print(bin_message)
len_message = len(ascii_message)

# preenche vetor freq_bin_message com a frequencia de cada elemento
freq_bin_message = np.zeros(len(bin_message))
for i in range(len(bin_message)):
    if bin_message[i] == '1':
        freq_bin_message[i] = fc1
    else:
        freq_bin_message[i] = fc2
```

Construindo a mensagem é transformado a mensagem em ascii 'hello comp' em binário tendo como saída:

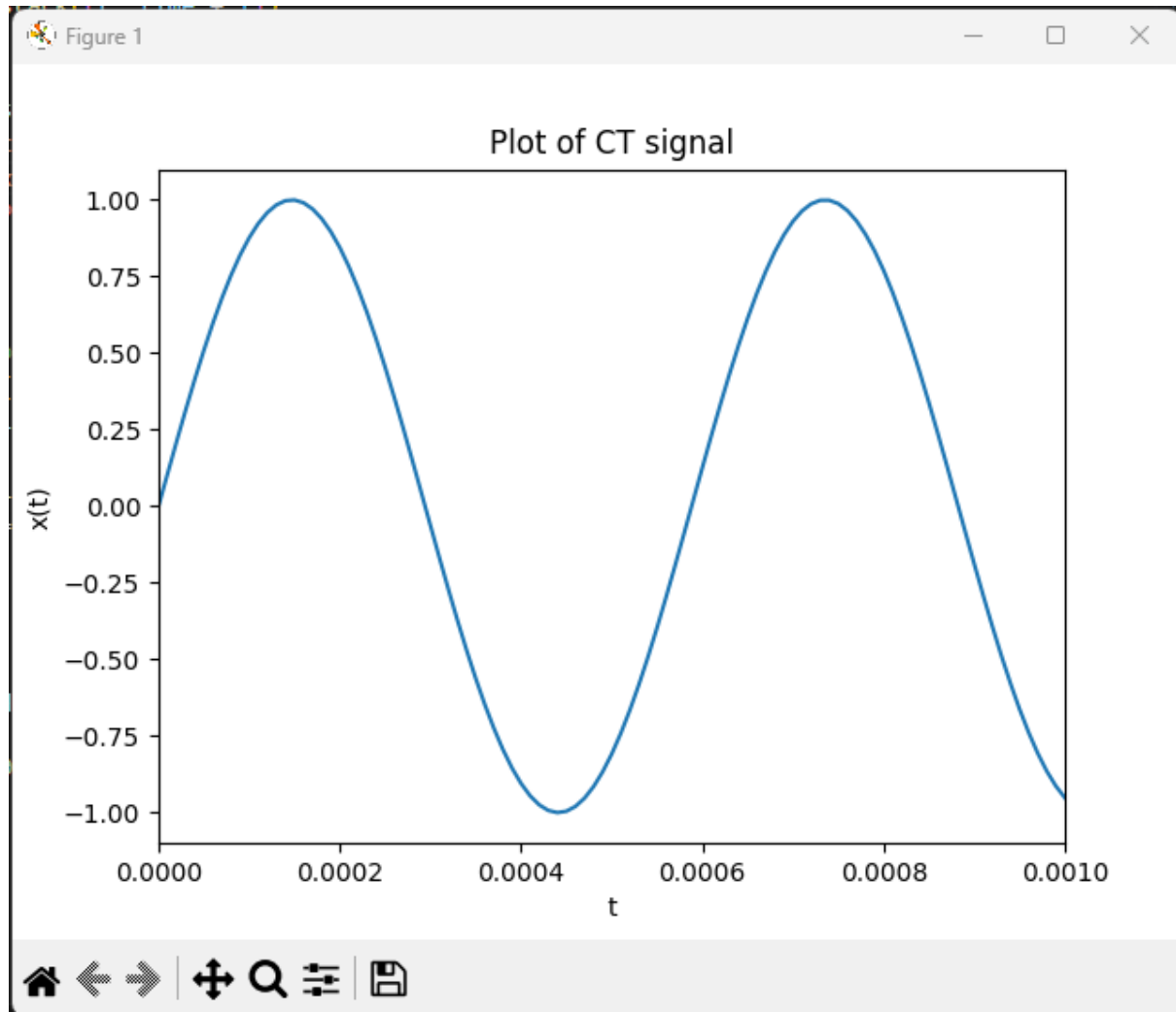
0b1101000011001010110110001101100011011110010000001100011011011110
110110101110000. Depois é armazenado o comprimento da mensagem ascii.

Após isso é feito um laço de repetição para preencher o vetor freq_bin_message com a frequência de cada elemento da mensagem se a mensagem, se o elemento binário da mensagem for 1 é utilizado a frequência Fc1 e não for 1, ou seja, 0 é utilizado a frequência Fc2.

```
# cria sinal modulado em frequência
signal = np.zeros(0)
t = np.zeros(0)
for i in range(len(freq_bin_message)):
    time, sin_signal = cont_sin(T, Fs, freq_bin_message[i])
    signal = np.hstack([signal, sin_signal])
    t = np.hstack([t, time + T])
    T += 0.1

plt.plot(t, signal)
plt.xlabel('t')
plt.ylabel('x(t)')
plt.title(r'Plot of CT signal')
plt.xlim([0, 0.001])
plt.show()
```

Na imagem anterior é a parte do código que representa o sinal modulado da frequência, onde logo após ele é plotado na tela como forma de gráfico onde podemos encontrar a saída:



Nesta imagem é criada o espectro do sinal, onde é colocado a variável T como período do sinal, utilizamos a diferença do segundo e do primeiro valor do Array "t". Na variável N recebe o tamanho do sinal já modulado.

Seguindo da função `fftfreq` que recebe como parâmetro o comprimento do sinal mais a variável do período T citada anteriormente.

```
# espectro do sinal
T = t[1] - t[0] # calcular o período do sinal 0.001 -> 1/T = 1000
N = signal.size

f = fftfreq(len(signal), T)
frequencias = f[:N // 2]
amplitudes = np.abs(fft(signal))[:N // 2] * 1 / N

print("Value in index ", np.argmax(amplitudes), " is %.2f" % amplitudes[np.argmax(amplitudes)])
print("Freq: ", frequencias[np.argmax(amplitudes)])
plt.plot(frequencias, amplitudes)
plt.grid()
plt.xlim([1000, 2000])
plt.show()
```

Em seguida é estipulada a amplitude para assim conseguir escrever na saída dos dados coletados anteriormente nas funções.

Por fim é plotado o gráfico da mensagem em frequência modularizada:

