



Instituto Nacional de Telecomunicações - Inatel Santa Rita do Sapucaí - MG – Brasil – Maio – 2024

Disciplina – TP547 – Princípios de Simulação

Resumo Crítico do Artigo com o Tema: Interface Gráfica baseado em Técnicas Avançadas de Modulação Usando Python

ALUNO: Daniel Malenga Moisés daniel.malenga@mtel.inatel.br

ALUNO: João Paulo Silva Dias joao.silva@dtel.inatel.br

Orientador: Prof. Dr. Samuel Baraldi Mafra

Resumo.

A modulação é um processo de modificar um sinal para transmitir informações através de um canal de comunicação.

É uma parte essencial dos sistemas de Telecomunicações modernos, e que desempenha um papel importante e crucial em comunicações com fio, sem fio, por satélite e ópticas.

O artigo em questão procurou demonstrar através de uma interface gráfica utilizando Python, as técnicas de modulação que podem ser amplamente classificadas em duas categorias principais:

Modulação Analógica e Modulação Digital.

Nesta estratégia os autores do artigo, exploram as diferentes técnicas de modulação usando interface gráfica e a linguagem de programação Python, com o principal objetivo de simular e analisar várias técnicas avançadas de modulação ASK, FSK, PSK.

- ASK- Modulação em Amplitude
- FSK- Modulação em Frequência
- PSK Modulação em Fase

I. INTRODUÇÃO

A modulação consiste no processo que envolve dois sinais, de variação das características do sinal da portadora de acordo com características do sinal de mensagem, ou seja, uma variação na **Amplitude, Frequência, Fase** de um deles implica em uma mudança correspondente no outro sinal.

O Sinal da mensagem é uma forma de onda periódica que contém as informações típicas para ser transmitido.

As técnicas de modulação são classificadas em dois tipos:

1. Técnicas de modulação analógica e
2. Técnicas de modulação digital.

O artigo propõe uma análise e simulação com interface gráfica da linguagem de programação Python para Técnicas de Modulação Digital.

Modulação Digital

As técnicas são ainda classificadas:

- ASK- Modulação em Amplitude
- FSK- Modulação em Frequência
- PSK Modulação em Fase

Em Modulação por Amplitude (ASK), o sinal de onda da portadora é classificado entre dois níveis, ou seja, binário **1** ou **0**, e ambos **os sinais da frequência e da fase da portadora** permanecem constantes quando ocorrem mudanças de amplitude, ou seja, mesmo que a amplitude do sinal varie, a frequência e a fase da portadora não são afetadas. Essa é uma característica e propriedade importantes em sistemas de Modulação em Amplitude (AM). Enquanto a Modulação em Frequência (FM), varia a frequência da portadora. Ambas as técnicas mantêm a fase e a frequência da portadora constante.

Para poder entender foi considerado um sinal retangular ou sinal quadrado como sinal de mensagem que precisa ser transmitido.

Portanto, a expressão matemática do sinal de mensagem foi expressa em:

$$\text{sig} = 1; t > 0 \\ 0; t \leq 0$$

A portadora é uma onda de rádio de alta frequência com amplitude (A_c) e frequência (F_c), representada por:

$$c(t) = A_c \cdot \cos(2\pi F_c t)$$

Por conseguinte, a equação para a modulação do sinal ASK foi expressa em:

$$\text{ASKsig} = \text{sig} * c(t)$$

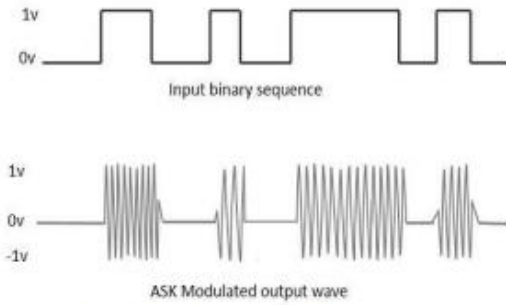


Fig 1 Input and Output waveforms

De forma similar em FSK, a frequência de onda da portadora é chaveada entre dois níveis, ou seja, binário 0 ou 1, isso de acordo com os dados binários de entrada.

O sinal FSK foi expresso no artigo matematicamente como,

$$FSKsig = \sin(2\pi f_1 t)$$

Onde, $f_1 = (F_c * sig)/2$

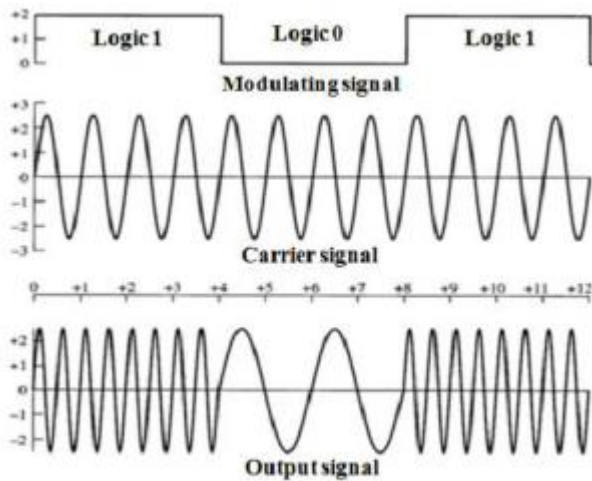


Fig2 Input and Output Waveforms of FSK

No PSK, a fase de portador é chaveada entre dois níveis de acordo com os dados recebidos da entrada, e a fase varia, enquanto a amplitude e a frequência permanecem constantes. As Figuras 1, 2 e 3 mostram as formas de onda de entrada e saída de Modulação ASK, FSK e PSK respectivamente.

Na PSK, a portadora sem mudar de fase é representada binário '1' e a fase portadora deslocada por 180°, binário '0'. PSK pode ser matematicamente expresso por:

$$PSKsig = \sin(2\pi F_c t + (\pi + \pi * sig/2))$$

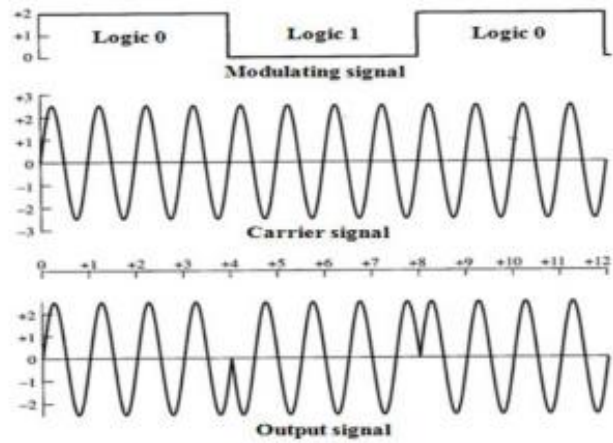


Fig 3 Input and Output Waveforms of PSK

II. OBRAS RELACIONADAS

A. Modulação Digital Usando Python.

Em seu livro com o título Modulação Digital usando Python, o autor Mathuranathan Viswanathan, descreve técnicas de simulação de modulação digital usando a linguagem Python, que é uma linguagem de código aberto, de alto nível com o propósito geral em programação, tendo um rico repertório com ampla variedade de bibliotecas e frameworks. Além de ser compatível com principais plataformas e sistemas operacionais.

O Python por ser uma linguagem de programação de código aberto, e de fácil interpretação, tem a vantagem de não ser necessário pagar pelas licenças de uso, tornando assim a escolha perfeita e vantajosa economicamente por ser gratuita tanto para desenvolvedores individuais como para empresas.

Python vem com uma variedade de pacotes como Scipy e NumPy, que suportam números complexos e computação científica. O Python é amplamente apreciado por uma grande comunidade global de programadores que desempenham um papel fundamental ao disponibilizar gratuitamente inúmeros pacotes. Isso facilita na realização de tarefas de desenvolvimento complexas.

O Python tem mantido o primeiro lugar no Ranking de escolha dos desenvolvedores e programadores de linguagem, desde 2017, do IEEE Spectrum.

B. Construção de Interface Gráfica (GUI) Usando Python com Tkinter.

Nesta etapa o autor Alan D. Moore, descreve a fase do desenvolvimento de interface gráfica do usuário utilizando a linguagem de programação Python. O Python oferece várias opções para o desenvolvimento de GUI (Graphical User Interface) e entre todos os métodos GUI, o Tkinter é o método mais comum e usado, tanto para desenvolvedores iniciantes como experientes. O Tk GUI é uma ferramenta fornecida com o Python, como a maneira mais rápida e fácil de criar aplicativos GUI tornando-se uma tarefa simples.

Etapas para criar um Aplicativo de Tkinter:

- 1° - Importando o módulo Tkinter.
- 2° - Criar a janela principal (contêiner).
- 3° - Adicionar qualquer número de widgets à janela principal.
- 4° - Aplicar o evento Trigger nos widgets.

Importar o módulo Tkinter é o mesmo que importar qualquer outro módulo no código Python. Ele é frequentemente escolhido para o desenvolvimento de aplicativos e ferramentas de desktop de pequeno a médio porte, porém possui suas limitações, particularmente em termos de estética moderna e recursos avançados.

Para projetos maiores e complexos, os desenvolvedores podem optar por outras bibliotecas GUI como PyQt, wxPython, que fornecem recursos mais avançados e uma gama mais ampla de elementos visuais.

No entanto, a simplicidade e a integração do Tkinter com Python continuam a torná-lo uma escolha preferida para fins educacionais e prototipagem rápida.

Tkinter é uma ferramenta de programação GUI em Python que oferece uma abordagem direta para a criação de aplicativos amigáveis. Sua compatibilidade entre plataformas, modelo orientado a eventos e integração com Python o tornam uma escolha ideal, permitindo que os desenvolvedores deem vida às suas ideias de software com facilidade e eficiência. À medida que o ecossistema Python continua a evoluir, o Tkinter continuará sendo um componente fundamental no mundo da programação GUI, preservando seu status como uma solução confiável e acessível para a construção de interfaces gráficas.

C. Construção de Interface Gráfica (GUI) Usando Python para modulação AM.

O autor Akhtar Nadaf nesta fase descreve sobre a aplicação de Modulação de Amplitude (AM) usando a interface GUI do Python.

Tkinter, numpy e matplotlib são as bibliotecas padrão Python, necessárias que devem ser importadas na criação de uma janela de Interface Gráfica com Usuário. Python & Tkinter é uma maneira rápida e fácil de criar aplicativos GUI (Graphical User Interface).

O **Tkinter** é uma ferramenta importante para criação de grande variedade de interfaces gráficas do usuário, incluindo janelas, caixas de diálogo e widgets personalizado, e adequado para criar aplicativos de desktop, adicionando uma interface gráfica ao programa de linha de comando.

O **NumPy** é um pacote de processamento de array de uso geral, que fornece um objeto de matriz multidimensional de alto desempenho e ferramentas para trabalhar com esses arrays. É o pacote fundamental para a computação científica com Python.

O **Matplotlib** é uma biblioteca de plotagem para a linguagem de programação Python e extensão matemática numérica do NumPy. Ele fornece uma API orientada a objeto para incorporar gráficos em aplicativos usando kits de ferramentas GUI de uso geral, como Tkinter, wxPython, Qt ou GTK.

III.MÉTODOS/SOFTWARE

Nesta fase o artigo descreve como foi desenvolvido e seus passos subsequentes, como também quais ferramentas utilizadas. Foi utilizado Python na versão 3.11 ou 3.10, com a importação das bibliotecas Matplotlib, e NumPy instaladas.

Com a finalidade de preparar o ambiente de simulação, o artigo descreve primeiramente a instalação do Visual Studio Code e em seguida, a instalação do Tkinter, Matplotlib e NumPy e realizando a importação dessas mesmas bibliotecas no código. Sugere-se também um arquivo novo e vazio com a extensão(.py), renomeado para Técnicas de Modulação Avançadas.

O Python é preferencialmente usado aqui ao em vez de Matlab, para simular técnicas de modulação, embora o último é tradicionalmente usado para este fim. Python revolucionou a computação científica e a visualização de dados com suas bibliotecas, tornando-se uma ferramenta valiosa para simular **Técnicas de Modulação**.

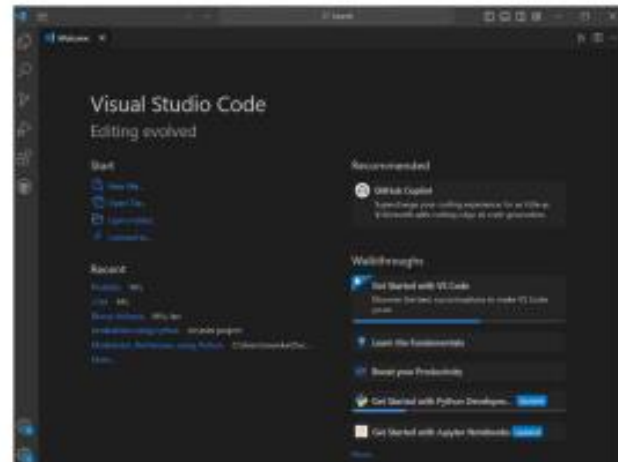


Fig. 04 - Interface de código do Visual Studio

O artigo usa funções de bibliotecas Tkinter, Matplotlib, scipy, numpy. Para a importação e instalação das bibliotecas, sugere-se instalar as bibliotecas antes de importar e para fazê-lo, use-se o comando pip install.

Exemplo: pip install matplotlib, para instalar matplotlib library. Primeiro importar vários pacotes de biblioteca Python no código, para operações suaves de funções.

```
from tkinter import *
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg, NavigationToolbar2Tk)
from tkinter import ttk
import numpy as np
import matplotlib.pyplot as plt
from math import pi
from scipy.fftpack import fft
from scipy import signal
```

Fig. 05 - Importando bibliotecas

Fig. 5 mostra como importar as bibliotecas e pacotes Python necessários para o projeto.

E em seguida, cria uma janela principal usando a função Tk() onde será adicionado outros componentes.

É também possível nomear um título para a janela principal

```
root = Tk()
root.title('Advance Modulation Techniques')
fig1 = Figure(figsize=(2,2) , dpi=100)

root.mainloop()
```

Fig. 06 - Criando janela Principal (Raiz)

Fig 6, demonstra a criação de janela usando a biblioteca Tkinter. Em seguida, constrói-se a interface figure-Tkinter-Canvas e organiza-se os widgets usando a função frame.

Use-se a classe FigureCanvasTkAgg para criar uma interface. O comando Tk()title é atribuído, como mostrado na Fig 6, para criar uma janela. Use-se o widget de entrada do pacote Tkinter para gerar o sinal da Portadora usando os valores de amplitude e frequência configurados pelo usuário.

IV. RESULTADOS E DISCUSSÃO

A Implementação

A fig 7 abaixo ilustra sobre o processo de fluxograma. Em primeiro lugar, o sinal de onda quadrada é gerado clicando no botão de plotagem dentro do widget. Em seguida, o usuário precisa inserir os valores de Amplitude e Frequência no widget de entrada para geração do sinal portador, que é usado para o processo de modulação.

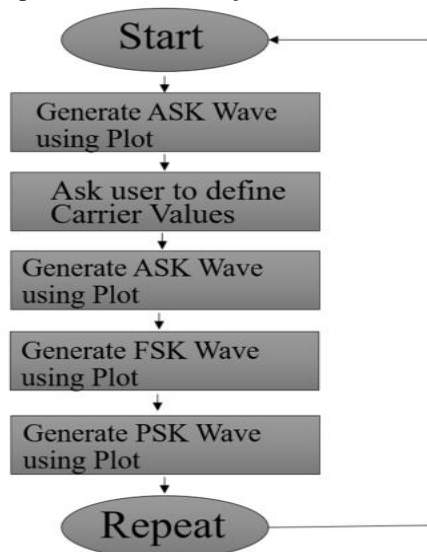


Fig. 07 - Fluxograma

Usando o botão de plotagem nos widgets do gerador ASK, FSK e PSK, as formas de onda ASK, FSK e PSK são geradas

e plotadas e para recomencar é necessário limpar toda a interface e regenerar as ondas usando o botão *clear*.



Fig. 08 - Interface GUI para Técnicas de Modulação Avancada.

Fig 8, é a janela gerada, e os quadros geradores de sinal consistem no botão de plotagem que é usado para plotar os sinais gerados.

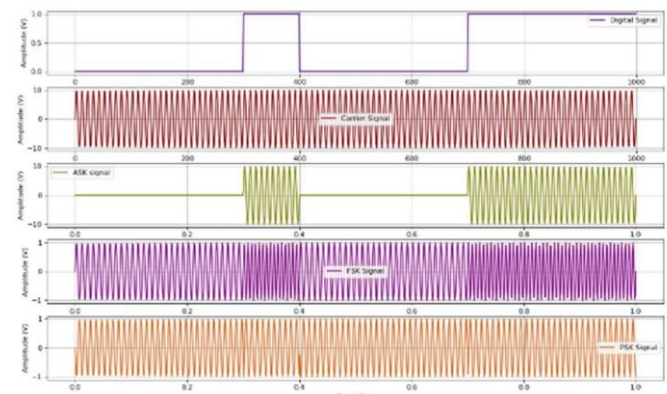


Fig. 09 - Forma de onda de saída 1

A Fig. 9 mostra as formas de onda de saída das técnicas de modulação ASK, FSK, PSK para a sequência de entrada 0001000111.

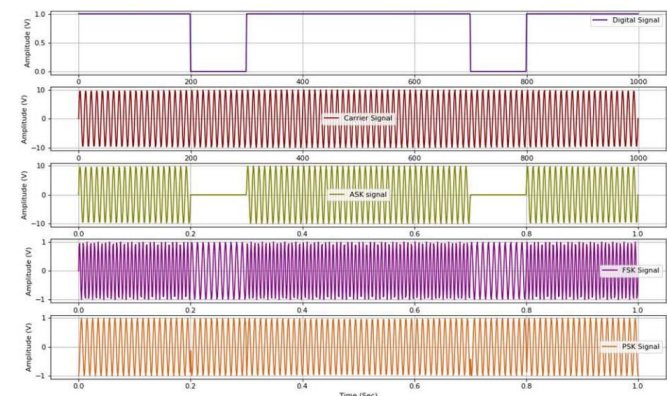


Fig. 10 - Forma de onda de saída 2

Fig. 10 mostra as formas de onda de saída das técnicas de modulações ASK, FSK, PSK para a sequência de entrada 1101111011.

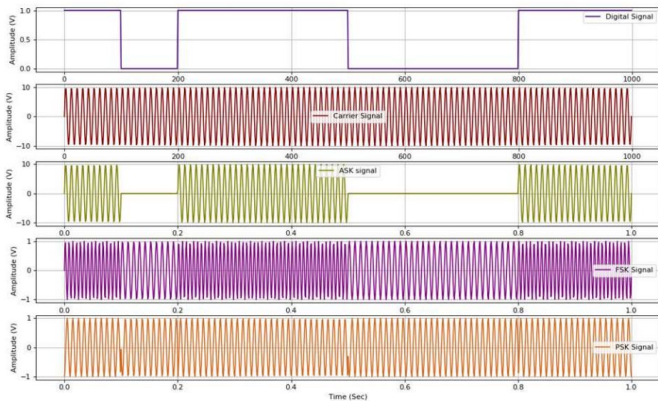


Fig. 11 - Forma de onda de saída 3

Fig. 11 mostra as formas de onda de saída das técnicas de modulação ASK, FSK, PSK para a sequência de entrada 1011100011.

A diferença entre ASK, FSK e PSK é claramente observada e perceptível a partir dos resultados e análise das figuras. No ASK, a portadora é transmitida somente quando o sinal de entrada é alto, ou seja, binário 1.

No FSK, quando a sequência de entrada é alta, uma portadora alta é transmitida e quando a sequência de entrada é de baixa frequência, a portadora é transmitida. Ele supera a principal desvantagem do ASK, ou seja, a modulação ASK é muito suscetível à interface de ruído.

Enquanto no PSK, há mudança de fase na saída PSK sempre que a sequência de entrada muda de 1 para 0 e vice-versa. Em comparação com o ASK, o PSK é menos suscetível ao ruído e, em comparação com o FSK, o PSK requer menos largura de banda.

V. CONCLUSÃO

Neste artigo, os autores exploraram de forma abrangente, as técnicas de modulação usando Python, com foco principal na compreensão, implementação e análise das principais técnicas de modulação ASK, FSK e PSK. Os principais objetivos do artigo foram obter insights sobre os fundamentos da modulação, desenvolver habilidades práticas de codificação em Python e avaliar o desempenho de diferentes técnicas de modulação.

Podemos tirar conclusões com confiança com base em nossas descobertas como Compreensão Fundamental, Proficiência em Python, Avaliação de Desempenho e Exploração Futura. Em resumo, o artigo sobre técnicas de modulação usando Python tem sido uma jornada gratificante. Alcançamos uma compreensão mais profunda da modulação, aprimoramos nossas habilidades de programação e contribuimos com insights valiosos sobre o desempenho de diferentes esquemas de modulação. Este artigo nos equipou com o conhecimento e as ferramentas para nos destacarmos no campo da engenharia de comunicação e abre caminho para futuras pesquisas e inovações.

VI. ÂMBITO DE APLICAÇÃO FUTURA

Processo de Modulação, aspecto fundamental dos modernos sistemas de comunicação, possibilitando a transmissão de informações através dos diversos canais de comunicação. Essas técnicas evoluíram significativamente ao longo dos anos, e seu futuro reserva grande promessa e potencial. O futuro das técnicas de modulação está intimamente interligado com as tecnologias emergentes, e várias áreas-chave prometem moldar seu desenvolvimento, principalmente, Comunicação Tera-hertz, Comunicação Quântica, 5G e além. O futuro das técnicas de modulação digital é empolgante e promissor, com potencial para revolucionar a comunicação de várias maneiras. Da comunicação quântica à tecnologia 5G e além, esses avanços continuarão a moldar nosso mundo interconectado, fornecendo soluções de comunicação mais rápidas, seguras e energeticamente eficientes. À medida que avançamos enfrentar os desafios e abraçar as oportunidades neste campo será crucial para realizar todo o potencial das técnicas de modulação digital nos próximos anos.

VII. Considerações Finais e Reprodução do Código

- O artigo apresenta uma contribuição relevante e equacionamentos são bem explicados pelos autores
- O artigo aborda um tema atual e pertinente para pesquisa, e contribui á ciência principalmente a comunicação quântica e 5G como também a inovação de futuras tecnologias como 6G.
- O artigo possui boa fundamentação quanto à escolha e justificativa no uso das ferramentas de pré-requisitos para as funcionalidades e simulação dos resultados usando GUI do Python.
- Uma abordagem abrangente e básica sobre as principais técnicas de modulação.
- A metodologia e apresentação dos resultados são claros e contribuem para a comunidade estudantil como também pesquisadores.

Reprodução dos resultados do artigo Modulação Digital com Sinal de Entrada: 0001000111 e seu Código.

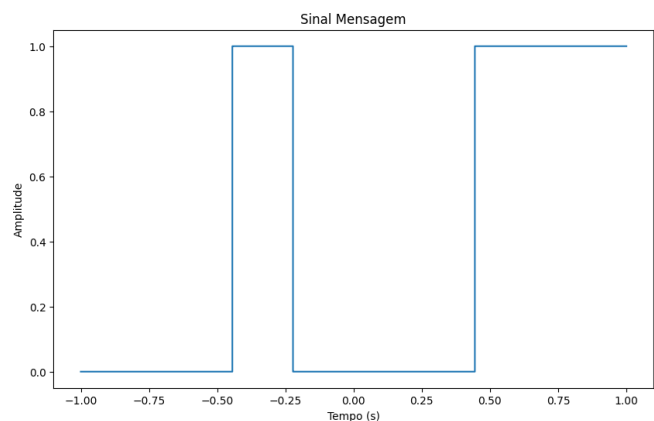


Fig. 12 – Sinal Mensagem Binária

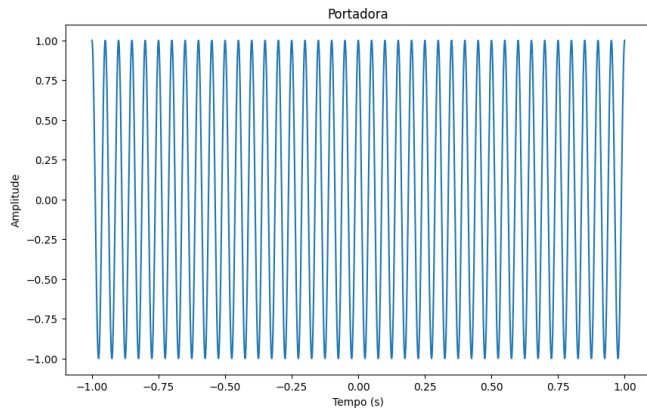


Fig. 13 – Sinal de Portadora

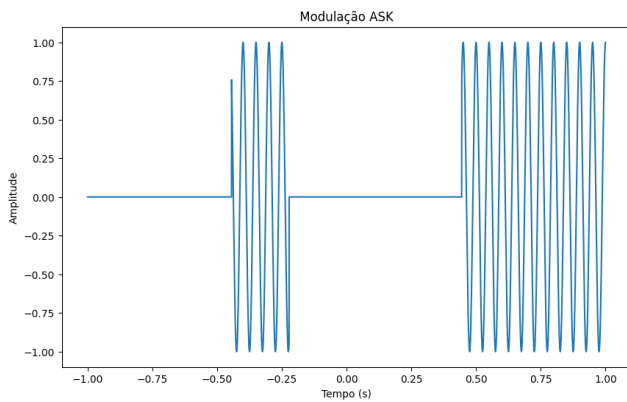


Fig. 14 – Sinal de Modulação ASK

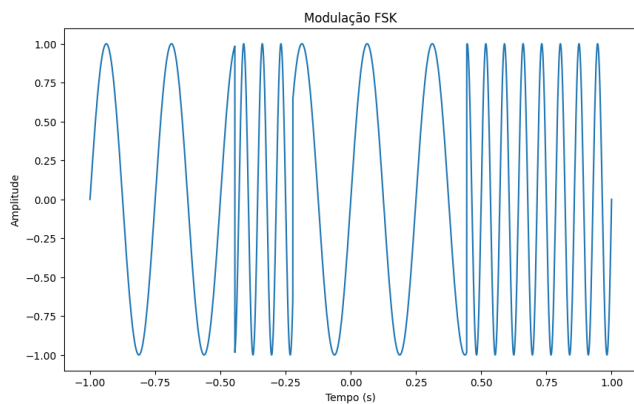


Fig. 15 – Sinal de Modulação FSK

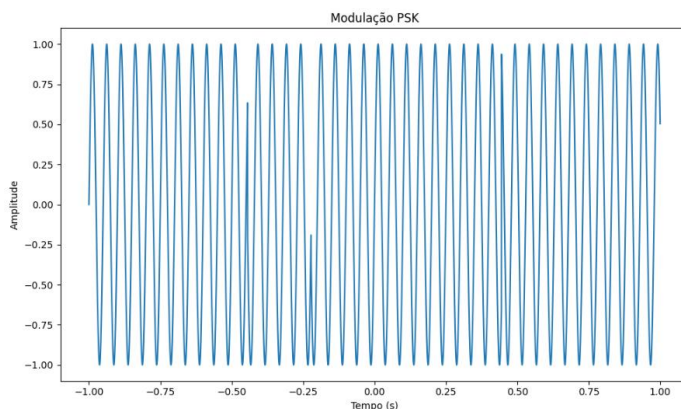


Fig. 16 – Sinal de Modulação PSK

```

!pip install matlab
!pip install numpy
!pip install matplotlib
!pip install scipy
!pip install random
import numpy as np
import matplotlib.pyplot as plt
import random
import pandas as pd
from scipy.special import erfc
from scipy import special
from scipy.fft import fft, fftfreq,
ifft
from scipy.signal import firwin,
lfilter
from scipy.io.wavfile import write
from scipy.signal import welch
from scipy import signal

#Parametros iniciais
N = 10000 # Números de amostras
t = np.linspace(-1, 1, N) # Tempo
#Portadora
A = 1 # amplitude
F = 20 # frequencia

sinal_entrada = "0001000111"
sinal_entrada = [int(bit) for bit in
sinal_entrada]
sinal_mensagem = np.interp(t,
np.linspace(-1, 1, len(sinal_entrada)),
sinal_entrada)
sinal_mensagem =
np.round(sinal_mensagem)

sinal_portadora
= A*np.cos(2*np.pi*F*t)
# ASK modulation function
modu_ASK
= sinal_mensagem*sinal_portadora

# FSK modulation function
f1 = 0.2*F + F*(sinal_mensagem/2)
modu_FSK = np.sin(2*np.pi*f1* t)

```

```

# PSK modulation function
fase = np.pi*(sinal_mensagem/2)
modu_PSK = np.sin(2*np.pi*F*(t + fase))

plt.figure(figsize=(10, 6))

plt.figure(1)
plt.plot(t,
sinal_mensagem,drawstyle='steps-post')
plt.title('Sinal Mensagem')
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')

plt.figure(figsize=(10, 6))
plt.figure(2)
plt.plot(t, sinal_portadora)
plt.title('Portadora')
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')

plt.figure(figsize=(10, 6))
plt.figure(3)
plt.plot(t, modu_ASK)
plt.title('Modulação ASK ')
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')

plt.figure(figsize=(10, 6))
plt.figure(4)
plt.plot(t, modu_FSK)
plt.title('Modulação FSK')
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')

plt.figure(figsize=(10, 6))
plt.figure(5)
plt.plot(t, modu_PSK)
plt.title('Modulação PSK')
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')

plt.tight_layout()
plt.show()

```

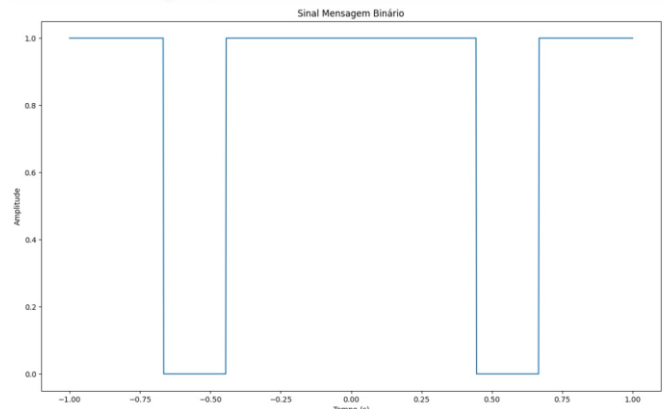


Fig. 17 – Sinal Mensagem Binária

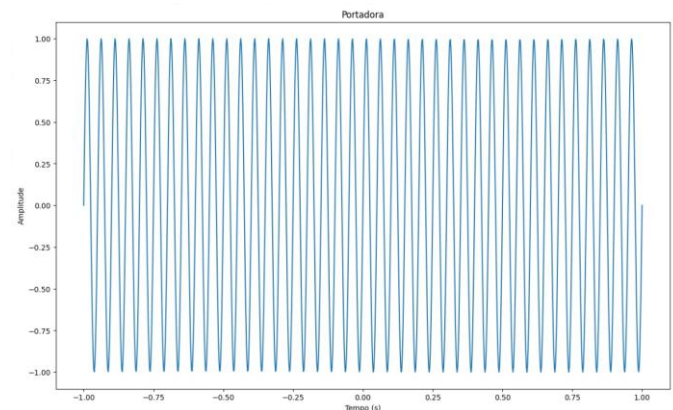


Fig. 18 – Sinal de Portadora

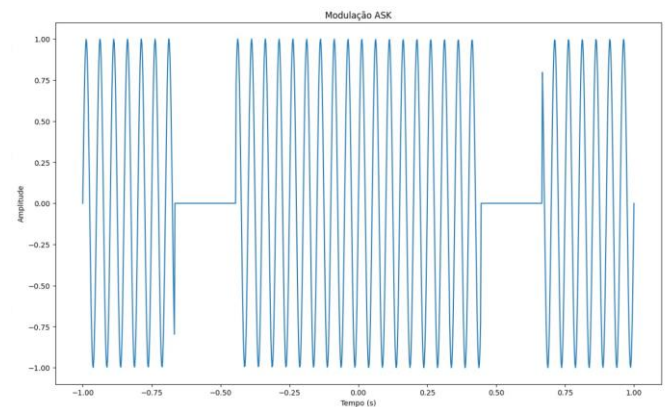


Fig. 19 – Sinal de Modulação ASK

Reprodução dos resultados do artigo Modulação Digital com Sinal de Entrada: 1101111011e seu Código.

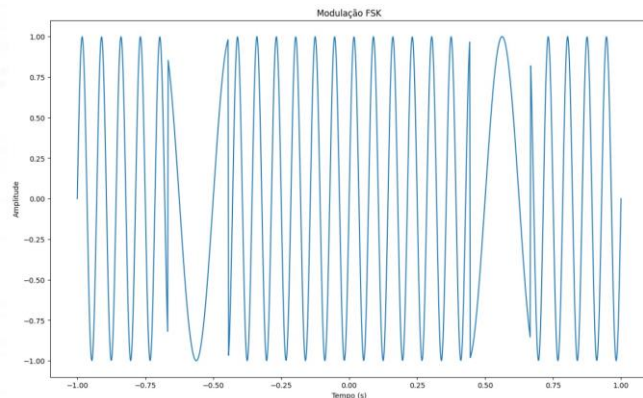


Fig. 20 – Sinal de Modulação FSK

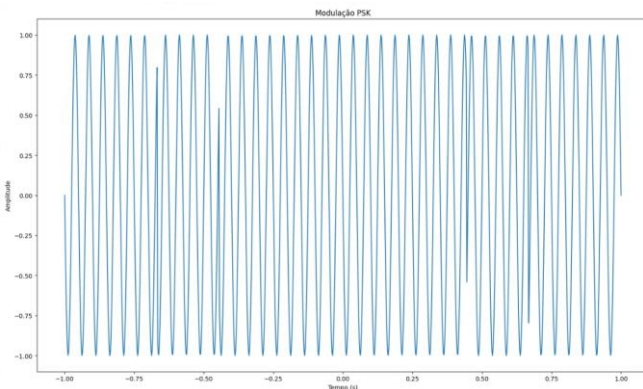


Fig. 21 – Sinal de Modulação PSK

```
!pip install numpy
!pip install matplotlib
!pip install scipy
!pip install random
import numpy as np
import matplotlib.pyplot as plt
import random
import tkinter as tk
import pandas as pd
from scipy.special import erfc
from matplotlib.backends.backend_tkagg
import FigureCanvasTkAgg,
NavigationToolbar2Tk
from scipy import special
from scipy.fft import fft, fftfreq,
ifft
from scipy.signal import firwin,
lfilter, welch
from scipy.fftpack import fft
from scipy.io.wavfile import write
from scipy import signal
from scipy.signal import hilbert
from ipywidgets import interact,
FloatSlider, Layout
```

```
N = 1000 # Número de amostras
t = np.linspace(-1, 1, N)

def modulacao(A, f_c, fase):
    Portadora = A * np.sin(2 * np.pi *
f_c * t)
    sinal_entrada = "1101111011"
    sinal_entrada = [int(bit) for bit
in sinal_entrada]
    sinal_mensagem = np.interp(t,
np.linspace(-1, 1, len(sinal_entrada)),
sinal_entrada)
    sinal_mensagem =
np.round(sinal_mensagem)
    modu_ASK = sinal_mensagem*Portadora
    f1 = 0.2*f_c + f_c*(sinal_mensagem
/ 2)
    modu_FSK = np.sin(2*np.pi*f1*t)
    modu_PSK = np.sin(2*np.pi*f_c*t +
fase*sinal_mensagem)
    plt.figure(figsize=(15, 9))
    plt.figure(1)
    plt.plot(t, sinal_mensagem)
    plt.title('Sinal Mensagem Binário')
    plt.xlabel('Tempo (s)')
    plt.ylabel('Amplitude')

    plt.figure(figsize=(15, 9))
    plt.figure(2)
    plt.plot(t, Portadora)
    plt.title('Portadora')
    plt.xlabel('Tempo (s)')
    plt.ylabel('Amplitude')

    plt.figure(figsize=(15, 9))
    plt.figure(3)
    plt.plot(t, modu_ASK)
    plt.title('Modulação ASK')
    plt.xlabel('Tempo (s)')
    plt.ylabel('Amplitude')

    plt.figure(figsize=(15, 9))
    plt.figure(4)
    plt.plot(t, modu_FSK)
    plt.title('Modulação FSK')
```



```

plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')

plt.figure(figsize=(15, 9))
plt.figure(5)
plt.plot(t, modu_PSK)
plt.title('Modulação PSK')
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')

plt.tight_layout()
plt.show()

# Ajustes para amplitude, frequência
portadora e fase
Amplitude = FloatSlider(min=1, max=2,
value=1, step=0.1, description='A:',
layout=Layout(width='50%'))

Frequencia = FloatSlider(min=20,
max=100, value=10, step=5,
description='F_c:',
layout=Layout(width='50%'))

phase = FloatSlider(min=(np.pi),
max=(2*np.pi), value=(np.pi), step=0.1,
description='Fase:',
layout=Layout(width='50%'))

interact(modulacao, A=Amplitude,
f_c=Frequencia, fase=phase);

```

Engineering and Applied Sciences, JANUARY 2, 2020, ISSN 1819-6608

- Subrato Bharati, Mohammad Atikur Rahman and Prajoy Podder “IMPLEMENTATION OF ASK, FSK AND PSK WITH BER VS. SNR COMPARISON OVER AWGN CHANNEL”, February, 2020, ResearchGate.
- Mohamed Wassim, “DIGITAL MODULATION ASK PSK USING PYTHON”, 4 August 2021, GitHub Repositories.
- Badrul Alam Mashfy, “DATA COMMUNICATION SIGNAL MODULATION”, 25 October 2021, GitHub Repositories.
- Alan D. Moore, “PYTHON GUI PROGRAMMING WITH TKINTER”, October 2021, Packt publishing.
- Akhtar Nadaf, “A PYTHON BASED GUI FOR AM MODULATION”, 21 February 2023, Electronics For You

REFERENCE

- G. Smithson, “INTRODUCTION TO DIGITAL MODULATION SCHEMES”, IEE Colloquium on The Design of Digital Cellular Handsets, August 6, 2002 Ref. No. 1998/240
- Shadbhawana Jain and Shailendra Yadav, “A SURVEY PAPER ON DIGITAL MODULATION TECHNIQUES”, JCSE International Journal of Computer Sciences and Engineering, December 31, 2019, E-ISSN: 2347-2693
- Mathuranathan Viswanathan, “DIGITAL MODULATIONS USING PYTHON”, December 2019, self-published book.
- Aaron Don M. Africa, John Joseph M. Castillo, Luis Anton D. Nieto and Shawn Reece T. Wu, “DESIGN OF AN ASK MODULATION DIGITAL SIGNAL CONVERSION SYSTEM” ARPN Journal of