

Universidade de Brasília
Departamento de Engenharia Elétrica

Trabalho final

Professor:
Paulo H. Portela de Carvalho

Introdução

O foco deste projeto final é o desenvolvimento de códigos de simulação para implementar várias modulações usando a abordagem de banda base complexa (que operam com base em símbolos)

No modelo de banda passante, que neste projeto será tratado como modelo de simulação de forma de onda, o sinal transmitido, o canal, o ruído e o sinal recebido são todos representados por amostras de formas de onda. Uma vez que cada detalhe da portadora RF é simulado, isto computacionalmente consome mais tempo e memória.

Representação de sinais

Por definição, um sinal banda passante é um sinal cujo espectro de energia de uma banda lateral está centrado em uma portadora de frequência f_c diferente de zero e não se estende a DC. Um sinal banda passante ou qualquer forma de onda de RF modulada digitalmente é representado como:

$$\tilde{s}(t) = a(t)\cos[2\pi f_c t + \phi(t)] = s_I(t)\cos(2\pi f_c t) - s_Q(t)\sin(2\pi f_c t)$$

onde

$$a(t) = \sqrt{s_I(t)^2 + s_Q(t)^2} \quad \text{e} \quad \phi(t) = \tan^{-1}\left(\frac{s_Q(t)}{s_I(t)}\right)$$

Notando que os termos seno e cosseno são componentes ortogonais, o sinal pode ser representado de forma complexa como:

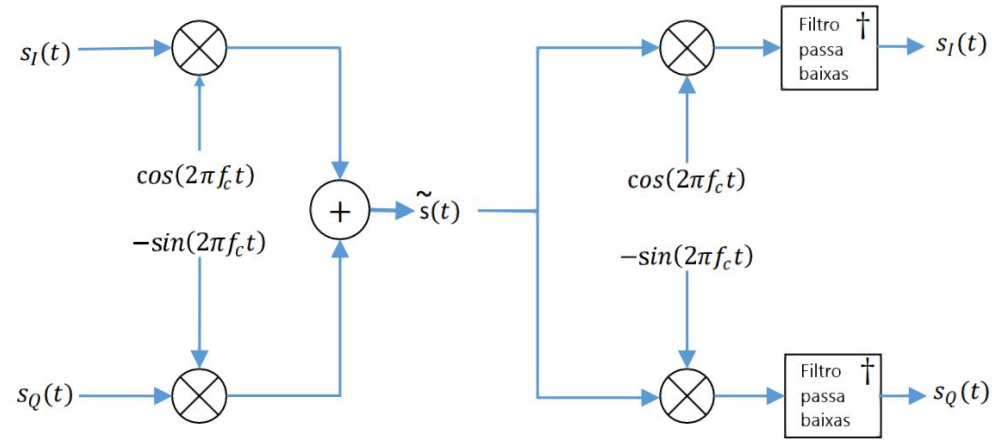
$$s(t) = s_I(t) + js_Q(t)$$

$$s(t) = s_I(t) + js_Q(t)$$

Nesta representação, o sinal $s(t)$ é chamado de envoltória complexa ou “representação equivalente em banda base complexa” do sinal real $\hat{s}(t)$.

As componentes $s_I(t)$ e $s_Q(t)$ são chamados de componente em fase e componente em quadratura. Comparando a equação anterior, identifica-se que na representação por envoltória complexa, a frequência da portadora é suprimida.

Em termos computacionais, isso reduz bastante a frequência de amostragem e as capacidades de memória necessária para simular o modelo.



*Componentes em banda base para
sinal banda passante*

*Sinal banda passante para
componentes em banda base*

Conversão prática de banda base para banda passante e vice-versa.
 Ferramental importante para as implementações computacionais
 exploradas neste projeto final

Implementando esquemas de modulação e demodulação em banda base complexa utilizando Programação orientada a objetos

A seguir descreve-se como as técnicas de modulação digital exploradas neste trabalho foram implementadas, em particular, o QAM, o FSK, o PSK e o PAM.

Conforme explicitamos anteriormente, todos os esquemas mencionados acima têm estrutura semelhante que pode ser explorada para escrever código simplificado usando programação orientada a objetos em Python.

Lógica implementacional

Começamos criando uma classe de base comum chamada ‘modulacao’. Em seguida implementamos as técnicas de modulação mencionadas como classes derivadas.

A função membro ‘modula’ da classe base ‘modulacao’ implementa a estrutura comum para as modulações mencionadas. A função simplesmente recebe os símbolos de informação de entrada e seleciona os símbolos modulados correspondentes de uma constelação de referência para a técnica de modulação escolhida.

A função é comum para as técnicas de modulação PAM, PSK, QAM e FSK. Portanto, as implementações da classe derivada para cada esquema, pode simplesmente herdar a implementação da classe base ‘modulacao’, sem a necessidade de implementação separada.

A função membro ‘demodula’ da classe base ‘modulacao’ implementa a estrutura comum para os demoduladores. A função, recebe como parâmetro os símbolos recebidos, realiza a detecção ideal usando um detector de IQ e retorna os símbolos detectados. O detector é comum para as técnicas PAM, PSK e QAM. Portanto, as implementações da classe derivada podem simplesmente herdar a implementação da classe base, sem a necessidade de implementação separada.

Apenas para o FSK, o método de detecção precisou ser substituído.

Lógica implementacional

```
class modulacao:
```

```
# Classe base: Modula  
# .M:          numero de pontos  
# .esquema:     esquema de modulação =PSK, QAM, PAM, FSK  
# .constelacao: constelação de referência
```

```
def modula(simbolos_entrada): #modula um vetor de símbolos de entrada usando o esquema de modulação escolhido
```

```
def demodula(simbolos_recebidos): #Demodula um vetor de símbolos recebidos usando o esquema de modulação  
escolhido
```

```
def iqDetector(simbolos_recebidos): # Detector ótimo para sinais
```

class PAM

class PSK

class FSK

class QAM

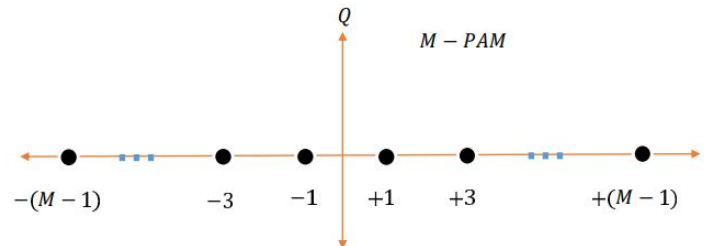
M-PAM

A modulação M-PAM é uma técnica de modulação unidimensional que não possui componente em quadratura ($s_Q = 0$). Todas as informações são codificadas na amplitude do sinal.

Um modelo banda base em tempo discreto para um modulador M-PAM, transmite uma série de símbolos de informação extraídos do conjunto com cada símbolo transmitido contendo k bits de informação ($k = \log_2(M)$). $m \in \{0, 1, \dots, M-1\}$. A expressão geral para gerar a constelação é dada por:

$$A_m = 2m + 1 - M, \quad m = 0, 1, \dots, M-1$$

Onde M denota a ordem de modulação e define o número de pontos na constelação de referência ideal. O valor de M depende de um parâmetro k (o número de bits que se deseja "sintetizar" em um único símbolo M-PAM. Por exemplo, se 3 bits ($k = 3$), isso resulta em uma configuração 8-PAM.



M-PSK

NO PSK, todas as informações são codificadas na fase do sinal da portadora. O modulador M-PSK transmite uma série de símbolos de informação extraídos do conjunto $m \in \{0, 1, \dots, M-1\}$. Cada símbolo transmitido contém k bits de informação ($k = \log_2(M)$). Os símbolos de informação são modulados usando o mapeamento M-PSK.

A expressão geral para gerar o conjunto de sinais M-PSK é dada por:

$$s_m(t) = A \cos \left[2\pi f_c t - 2\pi \frac{m}{M} \right], \quad m = 0, 1, \dots, M-1$$

Onde o parâmetro A é um fator de escala de amplitude

$$s_m(t) = A \cos \left[2\pi \frac{m}{M} \right] \cos(2\pi f_c t) + A \sin \left[2\pi \frac{m}{M} \right] \sin(2\pi f_c t), \quad m = 0, 1, \dots, M-1$$

que pode ser expresso como uma combinação de componentes em fase e em quadratura

$$s_m(t) = A \cos \left[2\pi \frac{m}{M} \right] + j A \sin \left[2\pi \frac{m}{M} \right], \quad m = 0, 1, \dots, M-1$$

M-QAM

Nas modulações M-QAM, os bits de informação são codificados como variações na amplitude e na fase do sinal. O modulador M-QAM transmite uma série de símbolos de informação extraídos do conjunto $m \in \{0, 1, \dots, M - 1\}$, com cada símbolo transmitido contendo k bits de informação ($k = \log_2(M)$).

Para restringir decisões errôneas por parte do receptor, os símbolos de informação são codificados via codificação Gray. Os símbolos de informação são então modulados digitalmente usando uma técnica retangular M-QAM, cujo conjunto de sinais é dado por

$$s = a + jb \quad \text{onde} \quad a, b \in \{\pm 1, \pm 3, \dots, \pm(\lceil \sqrt{M} \rceil - 1)\}$$

M-FSK

A modulação M-FSK pertence a uma classe mais ampla de modulação ortogonal, onde um sinal transmitido (correspondente a um símbolo fonte), é desenhado a partir de um conjunto de M formas de onda cujas frequências portadoras são ortogonais entre si.

M é o tamanho do alfabeto, de modo que cada símbolo representa $k = \log_2 M$ bits da informação fonte. Quando $M = 2$, o formato diminui para a modulação BFSK.

Este pequeno arcabouço teórico, inspirado pela obra *Communication Systems* de Simon Haykin (2017), direcionou as implementações em Python deste projeto.

Detector ideal no plano IQ usando distância Euclidiana mínima

Duas categorias principais de técnicas de detecção, comumente aplicadas para detectar os dados modulados digitalmente são a detecção coerente e detecção não coerente.

No modelo de simulação vetorial para a detecção coerente, o transmissor e o receptor concordam com a mesma constelação de referência para modular e demodular a informação. Os moduladores implementados neste projeto, incorporam o código para gerar a constelação de referência para o tipo de modulação selecionado. A mesma constelação de referência deve ser usada se a detecção coerente for selecionada como o método de demodulação.

Por outro lado, na detecção não coerente, o receptor está alheio à constelação de referência usada no transmissor. O receptor usa métodos como detecção de envelope para demodular os dados.

Na técnica de detecção IQ - um tipo de detecção coerente - a primeira etapa é calcular a distância Euclidiana par a par entre os dois vetores fornecidos: vetor de referência e símbolos recebidos corrompidos por ruído. Cada símbolo no vetor de símbolos recebido (representado em um plano p -dimensional) deve ser comparado com cada símbolo no vetor de referência. Em seguida, os símbolos, da matriz de referência, que fornecem a distância euclidiana mínima são retornados.

A distância Euclidiana entre dois pontos num espaço p -dimensional é dada por:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_p - y_p)^2}$$

Implementação computacional do Detector ótimo

```
def iqDetector (simbolos_recebidos):
```

O detector ideal calcula a distância Euclidiana par a par de cada ponto no vetor recebido contra cada ponto no vetor da constelação. O vetor `constelação'` representa os pontos ideais representados no plano IQ complexo.

Ele então retorna os símbolos decodificados que fornecem a distância Euclidiana mínima. Uma vez que este detector ideal pode ser usado para técnicas de modulação IQ como M-PSK, MQAM, M-PAM e o esquema de sinalização MFSK multidimensional, ele é codificado como uma função membro da Classe base `modula'`

```
from scipy.spatial.distance import cdist          #compute pair-wise Euclidean distances
```

documentado em: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html>

Roteiro de Testes

Existem 5 chamadas de funções ao final do código que são capazes de testar todos os módulos do aplicativo computacional. São elas:

- ❖ `stream(10, 16, 1)`
 - Responsável por gerar uma stream de palavras/símbolos m-ários equiprováveis ou não
 - O primeiro parâmetro (10) corresponde a quantos símbolos pretende-se gerar
 - O segundo, diz respeito ao tipo m-ário, podendo ser 8, 16 ou 64
 - O último parâmetro indica a equiprobabilidade (1 para equiprovável e 0 para não equiprovável)

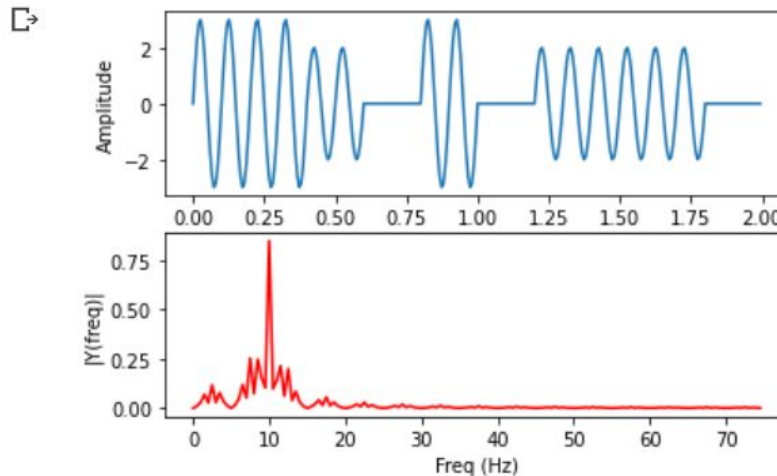
Exemplo de saída:

```
➡ [0 0 1 0]
   [1 0 1 1]
   [0 0 0 0]
   [0 0 1 0]
   [0 1 1 1]
   [0 0 1 0]
   [0 0 1 0]
   [1 1 1 0]
   [1 0 0 1]
   [0 1 1 1]
```

Roteiro de Testes

- ❖ `forma_onda(2, 16)`
 - Responsável por gerar a forma de onda de uma stream de palavras m-árias
 - O primeiro parâmetro corresponde ao tipo de modulação (1: PSK, 2:ASK, 3:FSK)
 - O segundo, diz respeito ao tipo m-ário, podendo ser 16 ou 64

Exemplo de saída:



Roteiro de Testes

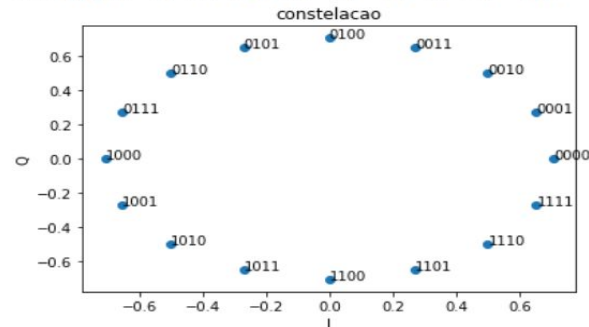
- ❖ `mod_demod(16, 1, 10)`
 - Responsável por modular uma stream de símbolos, gerar a constelação correspondente e demodular a stream modulada
 - O primeiro parâmetro (16) ao tipo m-ário, podendo ser (4, 8, 16, 64)
 - O segundo, diz respeito ao tipo de modulação (1: PSK, 2:QAM, 3:PAM e 4:FSK)
 - O último parâmetro indica o número de palavras para a modulação/demodulação

Exemplo de saída:

```

↳ (array([15,  6, 12, 13, 11,  3,  0,  1, 12,  9]),
   array([ 6.53281482e-01-0.27059805j, -5.00000000e-01+0.5j,
          -1.29893408e-16-0.70710678j,  2.70598050e-01-0.65328148j,
          -2.70598050e-01-0.65328148j,  2.70598050e-01+0.65328148j,
           7.07106781e-01+0.j,        6.53281482e-01+0.27059805j,
          -1.29893408e-16-0.70710678j, -6.53281482e-01-0.27059805j]),
   array([15,  6, 12, 13, 11,  3,  0,  1, 12,  9]))


```

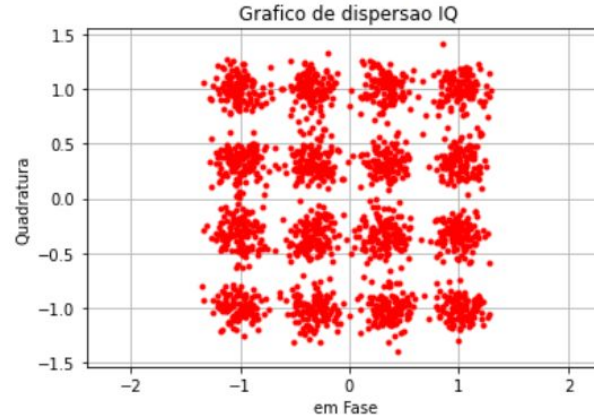


Roteiro de Testes

- ❖ `vetorial_ruido(1, 16, 10)`
 - Responsável por gerar a representação vetorial com ruído AWGN (nuvem de símbolos)
 - O primeiro parâmetro corresponde ao tipo de modulação (1: QAM, 2:PSK)
 - O segundo, diz respeito ao tipo m-ário, podendo ser 8, 16 ou 64
 - O terceiro, determina a relação EB/N0 em decibéis

Exemplo de saída:


`Eb/N0 = 10.00 dB and Es/N0 = 16.02 dB`
`BEP: Nbits = 400000, Nerror = 776, Pe_est = 1.940e-03`



Roteiro de Testes

- ❖ `SER(1)`
 - Responsável por calcular a Taxa de Erro de Símbolo (SER) e gerar o gráfico para vários tipos m-ários de um mesmo esquema de modulação
 - Esta função só tem um parâmetro que é o tipo de modulação que se deseja calcular a SER (1: PSK, 2: QAM, 3: PAM e 4:FSK)

Exemplo de saída:

