

SISTEMA DE COMUNICAÇÕES MÓVEIS
AP1 – parte 2 – TRABALHO EM GRUPO
PROF. CLAYTON JONES ALVES DA SILVA

Condições gerais:

1. O trabalho (parte 2 da AP1) perfaz 50% da nota da primeira avaliação bimestral.
2. O trabalho deve ser realizado e submetido em grupo. Os grupos são os mesmos designados para a disciplina.
3. A entrega do pedido será realizada por *commit* na plataforma do simulador do sistema de comunicações móveis, no repositório da plataforma do *GitHub*, em [claytonjasilva/simuladorSisCom \(github.com\)](https://github.com/claytonjasilva/simuladorSisCom).
4. Data de entrega do trabalho: 19 de outubro de 2022.

Dados do problema:

Seja um sistema digital que utiliza os caracteres alfanuméricos ASCII ([Tabela ASCII \(up.pt\)](#)). Sugestão: no Python, a função `ord()` recebe uma letra como parâmetro e retorna o ASCII.

O sistema permite elaborar mensagens que precisam ser enviadas através de um sistema de comunicações móveis, utilizando esquemas de modulação digital. As mensagens são enviadas em *frames* de 20 caracteres. Ou seja, 20 caracteres possuem 160 bits, os quais são codificados em símbolos de acordo com o esquema de modulação.

O sistema deverá utilizar um dos três esquemas de modulação: 8-PAM; 8-PSK; e 16-QAM. Lembrando que as expressões dos símbolos são dadas por:

M-PAM:

$$s_m(t) = A_m g(t) \cos(2\pi f_c t)$$

, onde $A_m = (2m - 1 - M) \cdot d$, $m = 1, \dots, M$ e d é a distância entre símbolos.

M-PSK:

$$s_m(t) = g(t) \cos \left[2\pi f_c t + \frac{2\pi}{M} (m - 1) \right]$$

, onde $m = 1, \dots, M$.

16-QAM:

$$s_m(t) = A_m g(t) \cos(2\pi f_c t + \phi_m)$$

, onde $A_m, \phi_n = [(2m - 1 - M) \cdot d, \frac{2\pi}{N} (n - 1)]$, $m = 1, \dots, M$, $n = 1, \dots, N$ e d é a distância entre símbolos.

Pedido:

1. Elaborar uma função do módulo *modulador* do simulador, nomeada *geraSimbolos*, que leia um *frame* de uma mensagem, ou seja, um conjunto de 20 caracteres e leia o esquema de modulação, dentre os indicados. A função deve gerar a sequência de bits do *frame*, de acordo com o código ASCII dos caracteres lidos, e retornar a sequência de símbolos a serem transmitidos.
2. Elaborar uma função do módulo *modulador* do simulador, nomeada *geraSinalPAM*, que leia uma sequência de símbolos 8-PAM. A função deve retornar o sinal modulado no

domínio do tempo e no domínio da frequência. Definir convenientemente os parâmetros necessários à simulação, como período de amostragem e intervalo de símbolos, entre outros. Admitir que $g(t) = 1$, no intervalo entre símbolos, ou seja, $g(t)$ é um pulso de onda quadrada.

3. Elaborar uma função do módulo *modulador* do simulador, nomeada *geraSinalPSK*, que leia uma sequência de símbolos 8-PSK. A função deve retornar o sinal modulado no domínio do tempo e no domínio da frequência. Definir convenientemente os parâmetros necessários à simulação, como período de amostragem e intervalo de símbolos, entre outros. Admitir que $g(t) = 1$, no intervalo entre símbolos, ou seja, $g(t)$ é um pulso de onda quadrada.

4. Elaborar uma função do módulo *modulador* do simulador, nomeada *geraSinalQAM*, que leia uma sequência de símbolos 16-QAM. A função deve retornar o sinal modulado no domínio do tempo e no domínio da frequência. Definir convenientemente os parâmetros necessários à simulação, como período de amostragem e intervalo de símbolos, entre outros. Admitir que $g(t) = 1$, no intervalo entre símbolos, ou seja, $g(t)$ é um pulso de onda quadrada.

5. Elaborar uma função do módulo *modulador* do simulador, nomeada *geraConstelacao*, que leia um sinal modulado por um esquema digital 8-PSK ou 16-QAM. Adicionar ao sinal um ruído gaussiano branco, um sinal de interferência ou perda de canal. Gerar o sinal resultante do efeito indesejado proporcionado pelo canal e plotar o sinal utilizando um mapeamento Gray.