

Simulando as Modulações FM e PM no Matlab

Para iniciar nossas simulações, de forma similar ao já feito para o sinal AM, criamos o sinal da onda portadora e o sinal com a mensagem que desejamos modular:

```
%% Criação da onda portadora:
Ac = 1;
fc = 50;
c = Ac * cos(2*pi*fc*t);

%% Criação do sinal de mensagem
fm = 10;
m = cos(2*pi*fm*t);
```

Em seguida, aplicamos as fórmulas da modulação, que são as seguintes:

Sinal PM:

$$A_c \cos[2\pi f_c t + k_p m(t)]$$

Para construir esse sinal no Matlab, iniciamos com um vetor vazio **s_pm** e começamos a preenchê-lo aplicando a fórmula para cada valor do vetor de tempo **t** e do sinal de mensagem **m**:

```
%% Criação do sinal modulado PM:
kp = 2;

% Criação de um vetor para guardar todos os valores obtidos
% Esse vetor vai começar com todos os valores zerados (só para
guardarmos
% um espaço na memória)
s_pm = zeros(1,length(t));

% Agora começamos a realmente preencher esse vetor aplicando a
fórmula da
% modulação PM em cada passo no tempo
for i = 1:length(t)
    s_pm(i) = Ac * cos(2*pi*fc*t(i) + kp*m(i));
end
```

Sinal FM:

$$A_c \cos \left[2\pi f_c t + 2\pi k_f \int_0^t m(\tau) d\tau \right]$$

Para o sinal FM podemos utilizar o mesmo método, mas dessa vez não utilizamos o sinal de mensagem **m**, mas sim a sua integral.

Para isso, escolhemos calcular essa integral numericamente com a função **cumtrapz** (função de cálculo numérico que já está inclusa no matlab outros métodos para calcular a integral também funcionariam).

Após calculada a integral do sinal, podemos calcular o sinal modulado de forma similar ao que fizemos para a modulação PM:

```
%% Criação do sinal modulado FM:
kf = 0.01;

% Integrar m_i
%{
A função cumtrapz calcula uma integral de forma numérica.
Precisamos de um sinal que representa a integral de m(t)

Especificamente, essa integral é calculada utilizando interpolação de
trapézios, mas isso é um assunto de cálculo numérico. Apenas vamos assumir
que esse método de integração dá um resultado "bom o suficiente".
%}
m_i = cumtrapz(m);

% Vamos agora calcular um valor de cada vez para o sinal modulado FM.
% Primeiro calculamos o coeficiente kp
kp = 2*pi*kf;

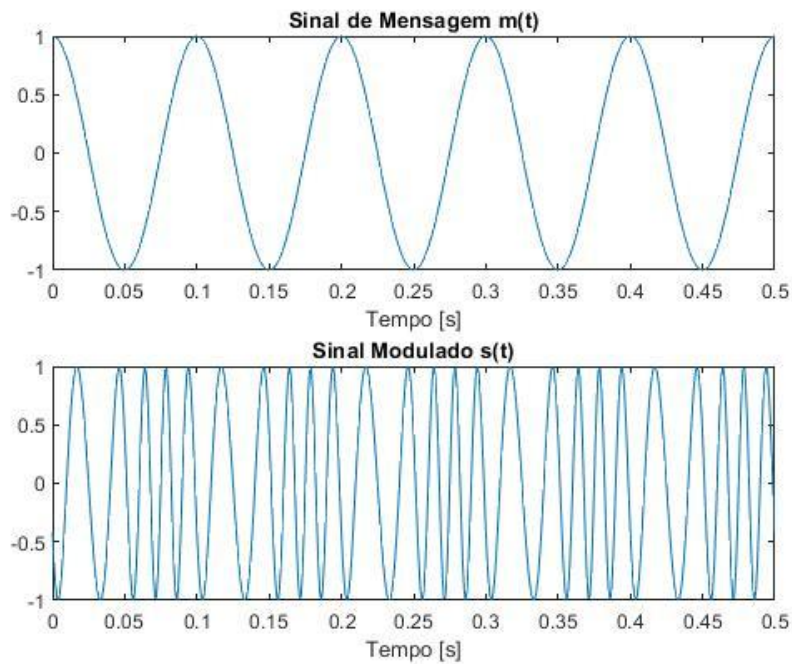
% Em seguida, vamos criar um vetor para guardar todos os valores obtidos
% Esse vetor vai começar com todos os valores zerados (só para guardarmos
% um espaço na memória)
s_fm = zeros(1,length(t));

% Agora começamos a realmente preencher esse vetor aplicando a fórmula da
% modulação FM em cada passo no tempo
for i = 1:length(t)
    s_fm(i) = Ac * cos(2*pi*fc*t(i) + kp*m_i(i));
end
```

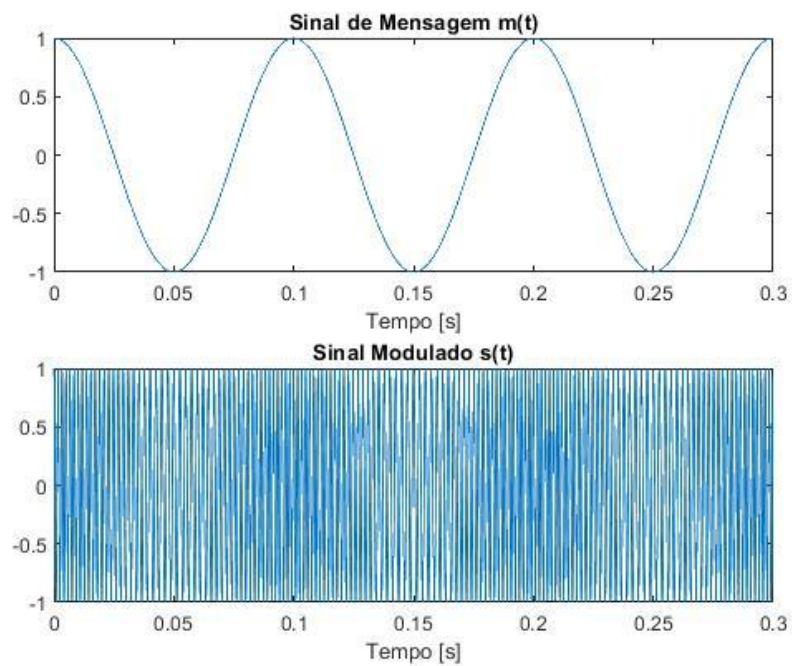
Resultados

A seguir, temos os gráficos dos sinais modulados em PM e FM. Observe que agora a informação transmitida está na “densidade” da onda portadora. Para o sinal PM, temos um gráfico mais “denso” quando o sinal de mensagem está crescendo, e para o sinal FM temos um gráfico “mais denso” quando o sinal de mensagem está em um pico.

Modulação PM:

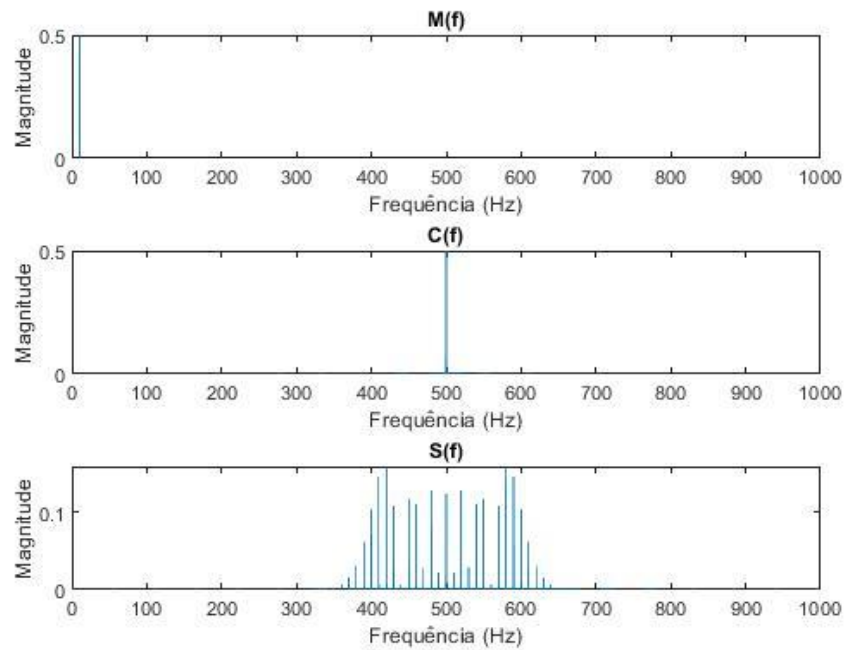


Modulação FM:

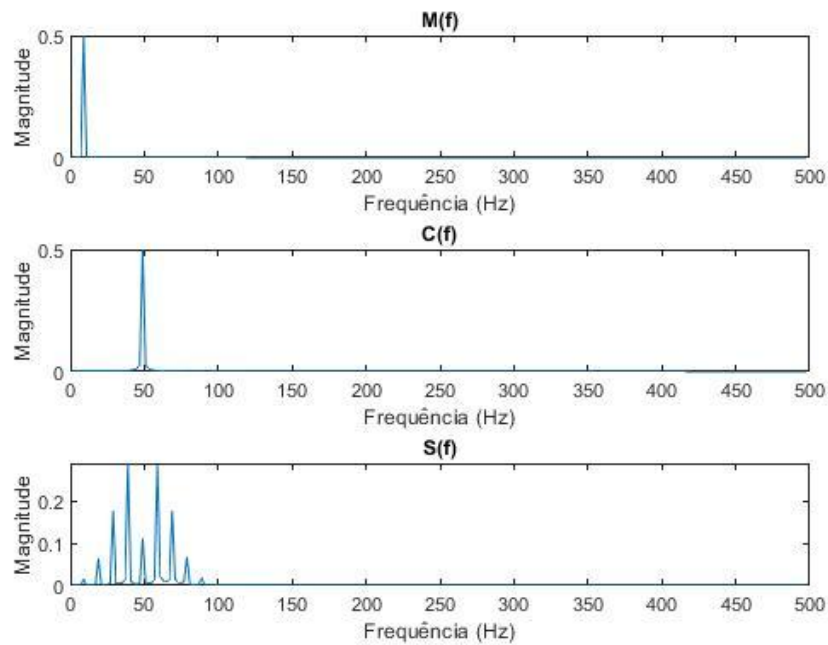


A seguir, temos também as transformadas de Fourier dos sinais PM e FM. Observe que agora o espectro é muito maior e imprevisível do que o do sinal AM.

Espectro do sinal modulado em PM



Espectro do sinal modulado em FM



Utilizando a função fmmod do pacote Communications Toolbox

Como mencionado anteriormente, o Matlab possui diversos pacotes que ajudam a facilitar simulações. Um deles é o **Communications Toolbox**, que contém funções para simulações de sistemas de comunicação. Para utilizá-lo, é necessário ter o pacote instalado no Matlab, algo que é configurado no momento da instalação.

Leia sobre o pacote e suas funções em:

<https://www.mathworks.com/help/comm/>

A seguir, fazemos a simulação da modulação de um sinal em FM. Observe como podemos simplesmente definir os parâmetros da modulação e ao final utilizar um único comando para fazer essa modulação, sem nos preocupar como a modulação em si é feita:

```
% Frequência de amostragem
fs = 2000;

% Frequência da portadora
fc = 500;
t = (0:1/fs:0.2)';

% Sinal de mensagem.
x = sin(2*pi*30*t) + 2*sin(2*pi*60*t);

% Desvio de frequência.
fDev = 50;

% Usar a função do communications toolbox.
y = fmmod(x, fc, fs, fDev);
```

