



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Centro Tecnológico
Departamento de Engenharia Elétrica

Princípios de Comunicações I COD: ELE8541

Prof. Responsável: Jair Silva (jair.silva@ufes.br)

Laboratório # IV – Modulação Angular

1. Objetivos

Nesta aula o aluno deverá simular os efeitos do Índice de Modulação (constante de modulação) angular e do Ruído AWGN na modulação FM (*Frequency Modulation*).

2. Conceitos Envolvidos

- Modulação de Angular FM
- Índice e Constante de Modulação
- Ruído Gaussiano Branco Aditivo (AWGN) e Relação Sinal-Ruído (SNR)

3. Experimento

Para a análise da Modulação de Frequência FM execute as seguintes tarefas no software de Computação Matlab:

- Estude o “script” **FM-PCI** apresentado em anexo para um bom entendimento dos processos de modulação e demodulação FM. Disserte sobre a envoltória e sobre a largura de banda ocupada pelo sinal modulado.
- Varie a Constante de Modulação β com valores próximos ao valor fornecido no script e observe a sua importância no processo como um todo. Analise o seu efeito nos domínios do tempo e da frequência.
- Simule o efeito de um canal que apenas adiciona ruído Gaussiano Branco nos sinais modulados usando o comando AWGN do Matlab. Varie o relação sinal ruído SNR em dB do sistema para o valor ótimo de Constante de Modulação. Calcule as potências dos sinais modulados.

4. Tarefa Extra-Classe

Insira o código que desenvolvestes para captura e processamento de um sinal de áudio e de uma imagem nos *script* de modulação FM criados e analise os efeitos do ruído nos mesmos. Para isto, compare o áudio e a imagem decodificados com os gerados para diversos valores de SNR.

5. Código AM-DSB-TC em Matlab

```
% ..... Princípios de Comunicações I.....
% Lab IV
% Modulação FM
% by Jair Silva
% UFES/2013
% .....
clc, clear all, close all;
% ..... Dados de entrada .....
fm = 440;           % Frequência do sinal modulante
Fc = 2000;          % Frequência da portadora
```



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Centro Tecnológico

Departamento de Engenharia Elétrica

```
fs = 50e3 ;           % Taxa de amostragem
A = 1;               % Amplitude do sinal modulante
B = 10000;           % Índice de modulação angular
C = 0;               % Fase inicial
d = 1 ;              % Duração do sinal de mensagem
%..... Dados calculados .....
kf = (B*fm)/A;        % Constante de modulação FM
delta_f = kf*A;        % Desvio frequencia -- Quando o sinal modulante é um tom
% ..... Geração do sinal modulante .....
t = linspace(0, d, d*fs+1); % sinal tempo
m = .5*cos(2*pi*fm*t); % sinal mensagem

% ..... Modulação FM .....
s_int = cumsum(m)/fs; % Computa a integração
s = A*cos(2*pi*Fc*t + B*s_int + C); % modulação

% ..... Canal .....
r = s; % B2B
% ..... Demodulação FM .....
% Executa a derivada do sinal modulado em FM
r_derv = diff(r);
% Filtragem butterworth filter parameter
N = 8 ; % ordem do filtro
fcut = Fc/2; % frequencia de corte do filtro
r_abs = abs(r_derv); % tira o valor absoluto do sinal modulado
[b,a] = butter(N,2*fcut/fs,'low'); % Filtro butterworth
figure, freqz(b,a,fcut,fs),title('frequency response of the LPF');
r_filt = filter(b,a,r_abs); % Filtragem
% Remove médias
r_sm = r_filt - mean(r_filt);
% Normaliza o sinal
r_demod = r_sm/max(abs(r_sm));
% ===== Plota alguns Gráficos =====
figure; % transmissão
f = linspace(-fs/2, fs/2, length(s));
S = fft(s);
subplot(2,1,1); plot(t,s); xlabel('t [s]'); ylabel('Sinal modulante');
subplot(2,1,2); plot(f,10*log10(fftshift(abs(S))));
xlabel('f [Hz]'); ylabel('Espectro do sinal modulante');
figure; % Receptor
R = fft(r_demod);
subplot(2,1,1); plot(t(1:end-1),r_demod); xlabel('t [s]'); ylabel('Sinal Demodulado');
subplot(2,1,2); plot(f(1:end-1),10*log10(fftshift(abs(R))));
xlabel('f [Hz]'); ylabel('Espectro do sinal Demodulado');
figure; % compara os sinais original e demodulado
subplot(2,1,1); plot(t,m,'b-',t(1:end-1),(-1)*r_demod,'r--')
subplot(2,1,2); plot(f,10*log10(fftshift(abs(fft(m)))),'b-',...
f(1:end-1),10*log10(fftshift(abs(R))),'r--');

% Para escutar os sinais
fprintf('Play do sinal original \n');
sound(m, fs);
fprintf('Play do sinal demodulado \n');
sound(r, fs);
```