

UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
ENGENHARIA DE COMPUTAÇÃO

Wagner Spinato Chittó

**4º TRABALHO PRÁTICO**  
Conversão analógico-digital  
Quantização não-linear

Santa Maria, RS  
19/09/2025

**Objetivo:**

Implementação de quantização não-linear (companding), criando um programa que realize a quantização não-linear usando a lei  $A$  ou lei  $\mu$ , e testar o programa implementado usando um sinal senoidal com faixa dinâmica de 40dB (ou sinal de áudio), gerando um gráfico da SQNR em função da faixa dinâmica.

## Procedimento:

### 1.1 - Criação do programa para o sinal senoidal:

```
clear all; close all;

mu = 255;                % Parâmetro de compressão (mu-law)
td = 0.002;              % Intervalo entre pontos do sinal "analógico"
t = td:td:1;             % 1 segundo de duração
amp = 1;                 % Amplitude de referência
ts = 0.02;               % Taxa de amostragem: 50 Hz
xsig = sin(2*pi*3*t);

faixa_dinamica_dB = 20:80; % Faixa dinâmica de 20dB a 80dB
SQNR_vec = zeros(size(faixa_dinamica_dB));

for i = 1:length(faixa_dinamica_dB)
    sig_scaled = xsig * 10^(faixa_dinamica_dB(i)/20);
    quantizado = quantizacao_mu(sig_scaled, mu);
    desquantizado = desquantizacao_mu(quantizado, mu);
    erro = sig_scaled - desquantizado;
    SQNR_vec(i) = 20 * log10(norm(sig_scaled) / norm(erro)); % SQNR
end

figure;
plot(faixa_dinamica_dB, SQNR_vec, 'bo-', 'LineWidth', 2);
xlabel('Faixa Dinâmica (dB)');
ylabel('SQNR (dB)');
title('SQNR em função da Faixa Dinâmica para Quantização Não-Linear');
grid on;

function y = quantizacao_mu(x, mu)
    % Compressão (q)
    y = sign(x) .* log(1 + mu * abs(x)) / log(1 + mu);
end

function x = desquantizacao_mu(y, mu)
    % Descompressão (dq)
    x = sign(y) .* ( (1 + mu) .^ abs(y) - 1 ) / mu;
end
```

## 1.2 - Adaptação do programa para um sinal de áudio ("StarWars3.wav"):

```
clear all; close all; clc;

[y, Fs] = audioread('./assets/StarWars3.wav');

y = double(y);
y = y / max(abs(y)); % normalizar para [-1,1]

mu = 255; % parâmetro
Nbits = 8; % número de bits
L = 2^Nbits; % níveis de quantização

faixa_dinamica_dB = 20:10:80;
SQNR_vec = zeros(size(faixa_dinamica_dB));

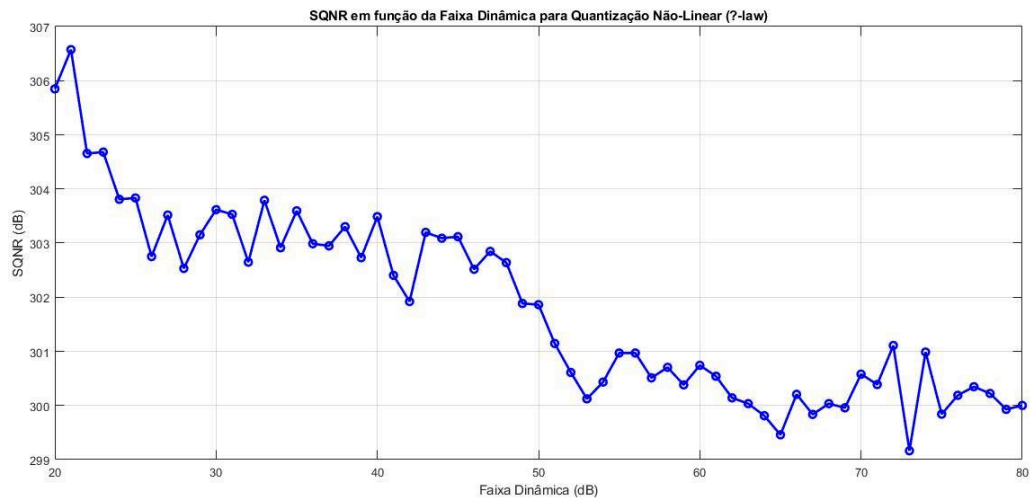
for i = 1:length(faixa_dinamica_dB)
    y_scaled = y * 10^(faixa_dinamica_dB(i)/20);
    y_mu = sign(y_scaled) .* log(1 + mu*abs(y_scaled)) / log(1 + mu);

    Delta = 2/L;
    qindex = floor((y_mu + 1)/Delta);
    qindex = min(qindex, L-1);
    yq_mu = -1 + Delta/2 + qindex*Delta;
    y_rec = sign(yq_mu) .* ( (1+mu).^abs(yq_mu) - 1 ) / mu;

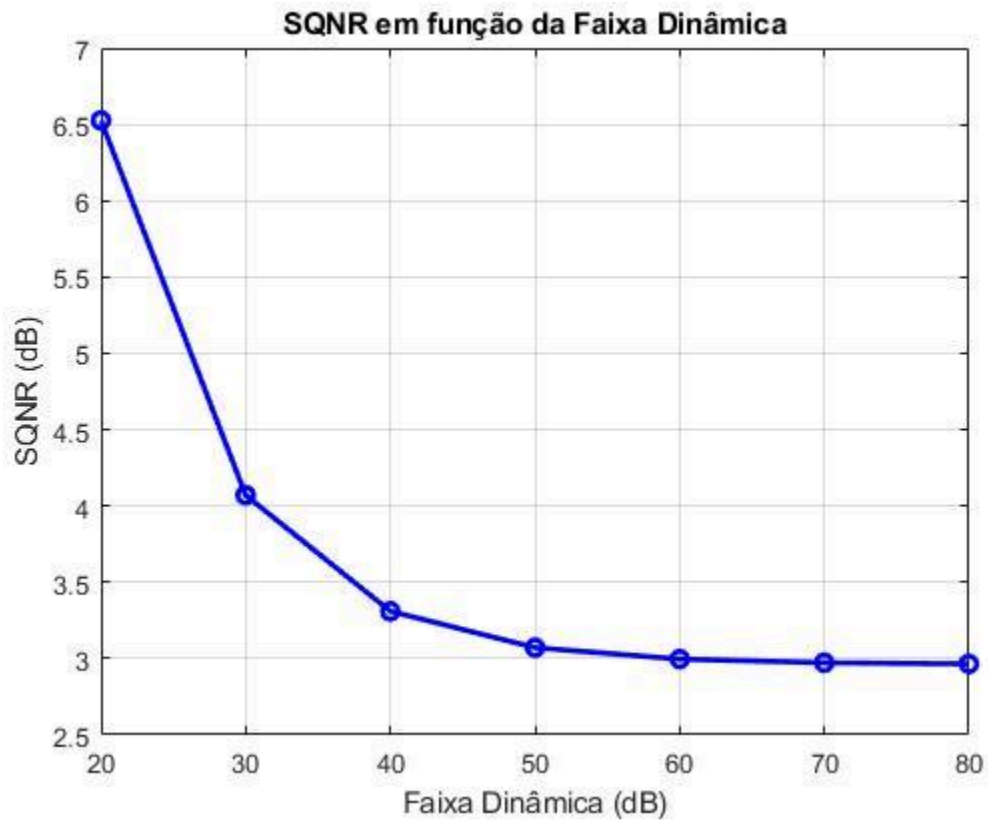
    % Calcular SQNR
    erro = y_scaled - y_rec;
    SQNR_vec(i) = 20*log10(norm(y_scaled)/norm(erro));
end

% Gráfico SQNR pela Faixa Dinâmica
figure;
plot(faixa_dinamica_dB, SQNR_vec, 'bo-', 'LineWidth', 2);
xlabel('Faixa Dinâmica (dB)');
ylabel('SQNR (dB)');
title('SQNR em função da Faixa Dinâmica');
grid on;
```

2.1 - Gráfico obtido para o sinal senoidal:



2.2 - Gráfico obtido para o arquivo “StarWars3.wav”



**Conclusão:**

Os resultados mostraram que a quantização não linear com a lei A melhora o SQNR em sinais com grande faixa dinâmica. Conforme a faixa dinâmica aumenta, o SQNR também cresce até o limite definido pelo número de bits do quantizador. A compressão antes da quantização faz melhor uso dos níveis disponíveis, preservando mais qualidade nos sinais de menor amplitude. Assim, o método de companding se mostrou mais eficiente do que a quantização linear em situações com baixa resolução.

**Bibliografia:**

- Slides da disciplina:
  - [https://ead06.proj.ufsm.br/pluginfile.php/5461157/mod\\_resource/content/3/AULA\\_7.pdf](https://ead06.proj.ufsm.br/pluginfile.php/5461157/mod_resource/content/3/AULA_7.pdf)
  - [https://ead06.proj.ufsm.br/pluginfile.php/5461159/mod\\_resource/content/2/Quadro%20branco%20%282021-12-01%29.pdf](https://ead06.proj.ufsm.br/pluginfile.php/5461159/mod_resource/content/2/Quadro%20branco%20%282021-12-01%29.pdf)