

AE5 - Modulação de Código de Pulso

Jessica de Souza

Engenharia de Telecomunicações, Instituto Federal de Santa Catarina

[<jessica.souzajds@gmail.com>](mailto:jessica.souzajds@gmail.com)

29 de maio de 2018

1. Introdução

O objetivo deste trabalho é aplicar os conhecimentos adquiridos sobre o tema de quantização de sinais. Para isto, foram realizadas algumas simulações de quantização de um sinal de áudio, juntamente com a transmissão binária deste através da sinalização NRZ unipolar com variação de SNR e também variação de bits de quantização. As próximas seções irão descrever o embasamento teórico utilizado para este trabalho e os resultados da simulação.

2. Conceitos Abordados

As próximas seções irão descrever o embasamento teórico utilizado para este trabalho e os resultados da simulação.

2.1. Modulação PCM

A modulação de código de pulso (PCM) é um método utilizado para realizar a representação digital de um sinal analógico. Basicamente, a modulação PCM possui três etapas: **amostragem**, **quantização** e **codificação** [1]. Na primeira etapa, que é a **amostragem**, são coletados os valores do sinal em um intervalo de tempo que é baseado em uma frequência de amostragem (F_s), respeitando sempre a teoria de Nyquist. Quanto maior for F_s , mais rico em detalhes será o sinal amostrado.

Em seguida é realizada a **quantização**, onde o sinal amostrado é nivelado e arredondado para o nível de tensão de referência mais próximo. Para realizar tal atividade é necessário saber o número de bits de quantização (l), que irá determinar o

número de níveis utilizado (L). A relação entre número de bits e número de níveis é $L = 2^n$ [2]. Após isso, utilizamos o L para saber quantos volts por nível será utilizado para quantizar, isso é determinado pelo passo-delta, que é a divisão do valor máximo do sinal pelo número de bits. Por fim, para realizar a quantização do sinal, basta dividi-lo pelo passo delta e depois arredondá-lo.

Por fim, na etapa de **codificação** é realizada a transformação do sinal de um formato decimal em volts para binário, onde este pode ser transmitido através de diversas técnicas de codificação existentes que são chamadas de técnicas PCM [2]. Neste trabalho, optou-se por utilizar a codificação não-retorna-à-zero (NRZ) unipolar, onde o sinal transmitido possui apenas dois níveis de tensão (-1 e +1), onde suas vantagens já foram descritas em trabalhos anteriores. A Figura 1 mostra os processos utilizados em uma modulação PCM.

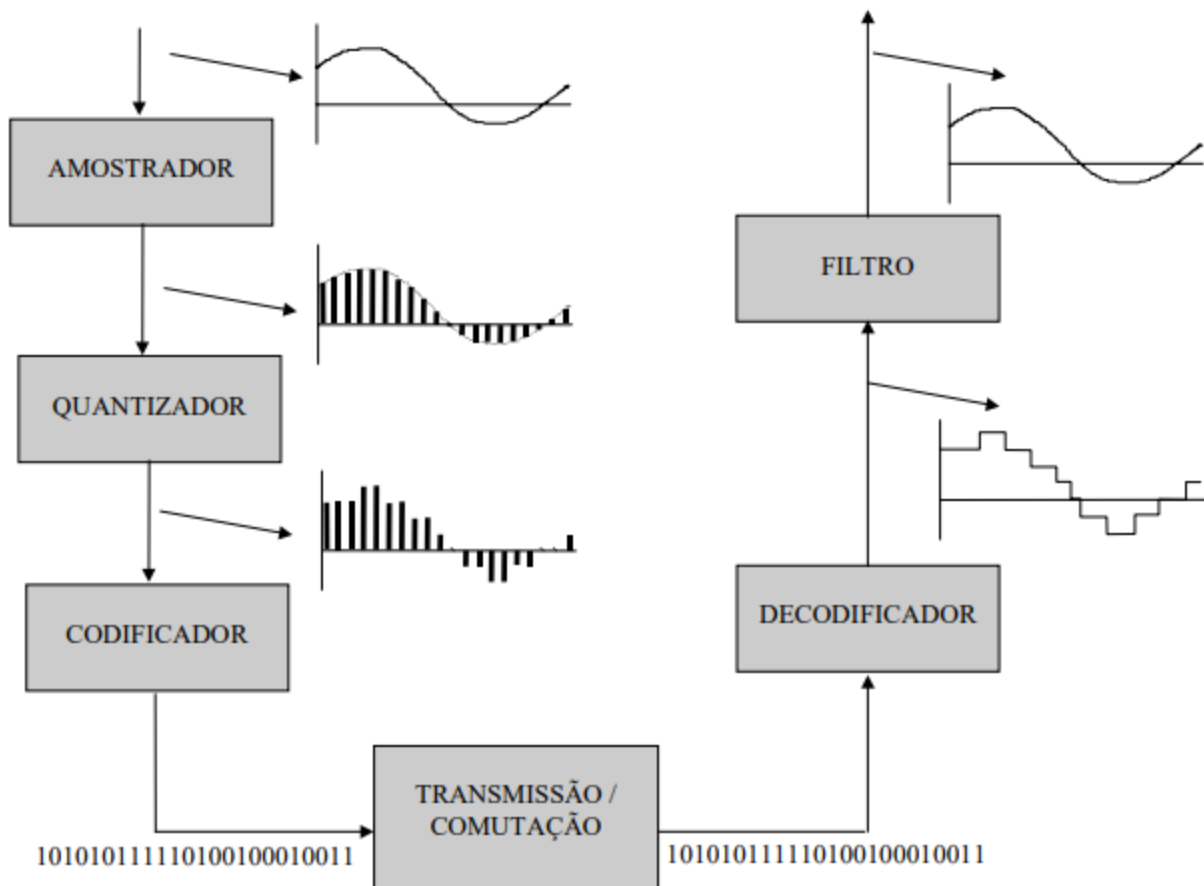


Figura 1 - Modulação PCM.

3. Metodologia

Os experimentos foram executados na ordem descrita na introdução. A seguir, o detalhamento de como foi realizado cada um deles. Foi utilizado o software Matlab para a realização dos experimentos e a toolbox “Data Acquisition” para captura do áudio utilizado no trabalho. Foram realizadas duas linhas de testes: a primeira consistiu em realizar o processo de quantização do sinal uniforme do sinal variando o número de bits de quantização e a segunda consistiu em manter o número de bits em um valor fixo e realizar a variação da SNR.

Utilizando a toolbox Data Acquisition do Matlab, foi realizada a captura de um sinal de voz com duração de 5 segundos, amplitude variando entre -1 e 1, e frequência de amostragem de 44100 Hz. Esta primeira etapa, foi realizada a variação do número de bits de quantização para os valores de: 3, 5, 8 e 13 bits. Para isso, foi utilizado um valor de SNR=100 para que não haja interferência de ruído nesta análise de como o número de bits interfere na qualidade do sinal. A seguir, o código que foi utilizado para realizar a extração do sinal, quantização e codificação. A fundamentação de quantização e codificação que foi utilizada neste trabalho está na seção 2.1.

```
1 % Extraindo características do sinal
2 [y,Fs] = audioread('captura_jessicasouza2.wav');
3 info_audio = audioinfo('captura_jessicasouza2.wav');
4 t = 0:1/Fs:5-1/Fs;
5 k = 8;          % Bits de quantizacao: variar entre 3, 5, 8 e 13
6 L = 2^k;        % Niveis de quantizacao
7
8 % Quantizacao
9 y_up_pos = y+1;          % Inserindo offset de 1v no audio
10 passo_delta = max(y_up_pos)/L; % Volts por nivel
11 y_q = y_up_pos/passo_delta; % Sinal quantizado
12
13 % Codificacao
14 y_dec = round(y_q);      % Transformando para decimal
15 y_bin = de2bi(y_dec);    % Transformando em binario para transmitir
16 [m,n] = size(y_bin);
17 y_dig = reshape(y_bin, 1, m*n); % Sinal em formato digital
```

Após realizar a quantização e codificação do sinal, foi realizada a simulação de transmissão do sinal binário em um canal AWGN utilizando NRZ unipolar. A seguir, o código utilizado que caracteriza esta seção.

```

18 % Transmitindo em um canal agwn uzando NRZ polar e filtro casado
19 N = 10;
20 y_dig2 = (y_dig*2)-1;
21
22 % Gerando o filtro
23 filtro_tx = ones(1,N);           % Pulso de N amostras
24 filtro_rx = filtro_tx;
25 info_up = upsample(y_dig2,N);    % Insere N amostras a cada valor da informacao
26
27 % Passando o sinal pelo filtro
28 y_tx = filter(filtro_tx,1,info_up); % Este é o sinal transmitido
29 t2 = [0:5/length(y_tx):5-1/length(y_tx)];

```

Uma vez transmitido, o sinal será recebido no canal AWGN, onde foi utilizada a SNR=100. Foi escolhido filtro casado para que o sinal tivesse um melhor desempenho. A seguir o código referente à esta parte. Nos resultados poderemos ver a diferença no tempo do sinal com e sem filtro casado.

```

30 % Recepcao do sinal - Canal AWGN
31 SNR = 100; % Variar SNR entre: 0, 5 e 10 para 8 niveis (fixo). Para variar os niveis mantem SNR
    =100
32 y_rx = awgn(y_tx,SNR);
33 y_rx_filt = filter(filtro_rx,1,y_rx)/N;
34 Z_t = y_rx_filt(N:N:end); % Pega o valor de bit no fim de cada pulso
35 y_fim = (Z_t > 0);

```

Por fim, foi realizada a decodificação do sinal (processo inverso ao realizado no início do experimento) para que se possa ouvir a mensagem recebida. A seguir o código desenvolvido para a decodificação do sinal.

```

36 % Decodificação e restauração do sinal recebido
37 y_dig_rx = reshape(y_fim, m, n);
38 y_int = bi2de(y_dig_rx);
39
40 y_int = y_int/L;           % Volta ao formato original
41 y_int = y_int - 1;        % Remove o offset
42 sound(y_int,Fs)

```

Na segunda etapa foi realizado o mesmo procedimento descrito anteriormente, onde foi realizada a variação da SNR com os valores: 0, 5 e 10, mantendo o número de bits de quantização em 8.

4. Resultados e discussão

Para a primeira parte do experimento que foi utilizada a variação do número de bits de quantização, obteve-se os resultados de acordo com as imagens a seguir.

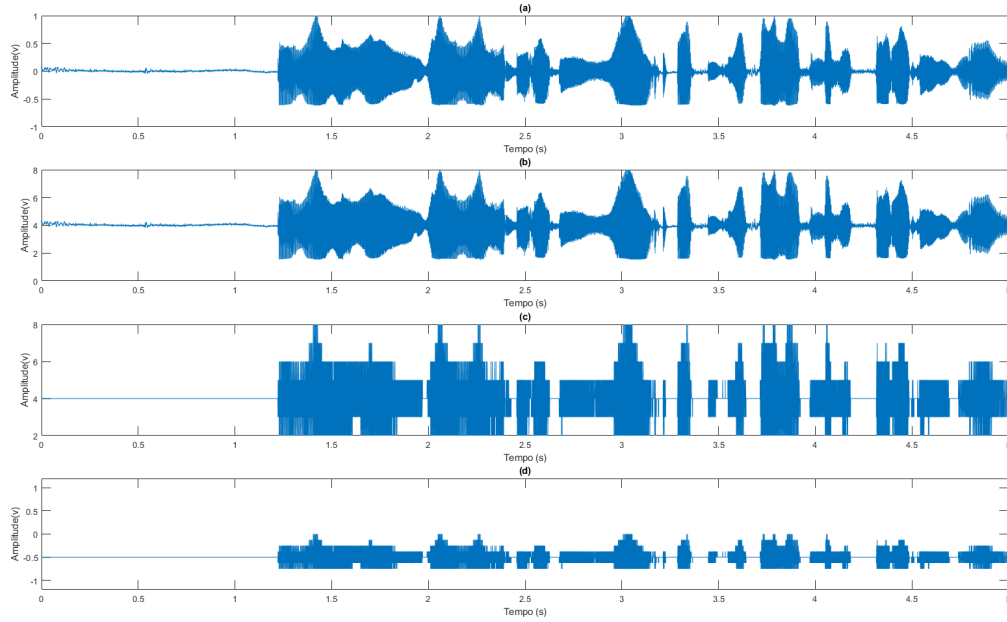


Figura 2 Setup 1 - SNR 100 e $k=3$ bits. (a) Sinal original, (b) Sinal quantizado, (c) Sinal codificado em decimal e (d) Sinal recebido.

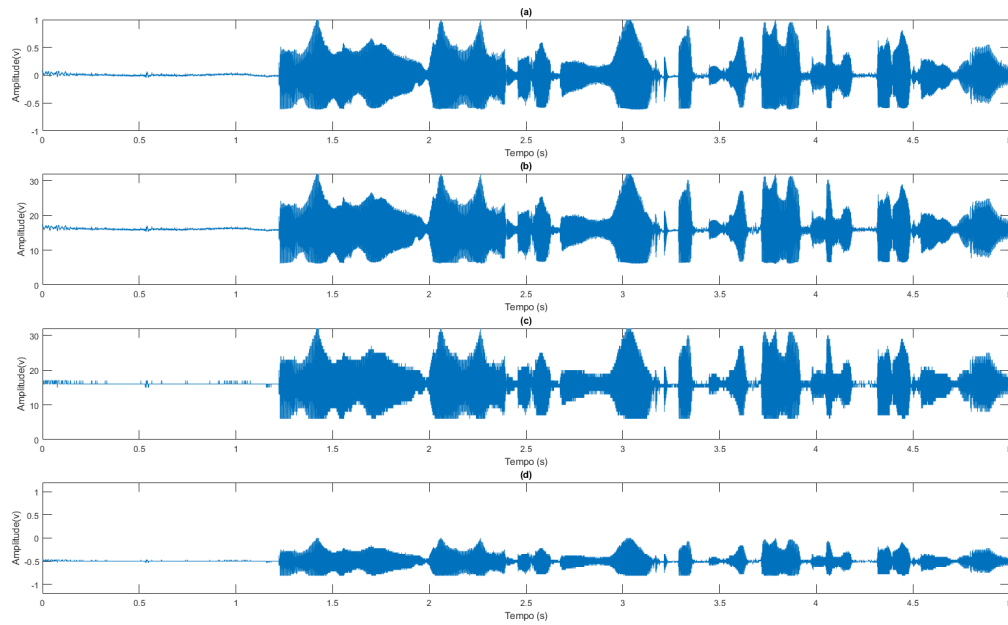


Figura 3 Setup 2 - SNR 100 e $k=5$ bits. (a) Sinal original, (b) Sinal quantizado, (c) Sinal codificado em decimal e (d) Sinal recebido.

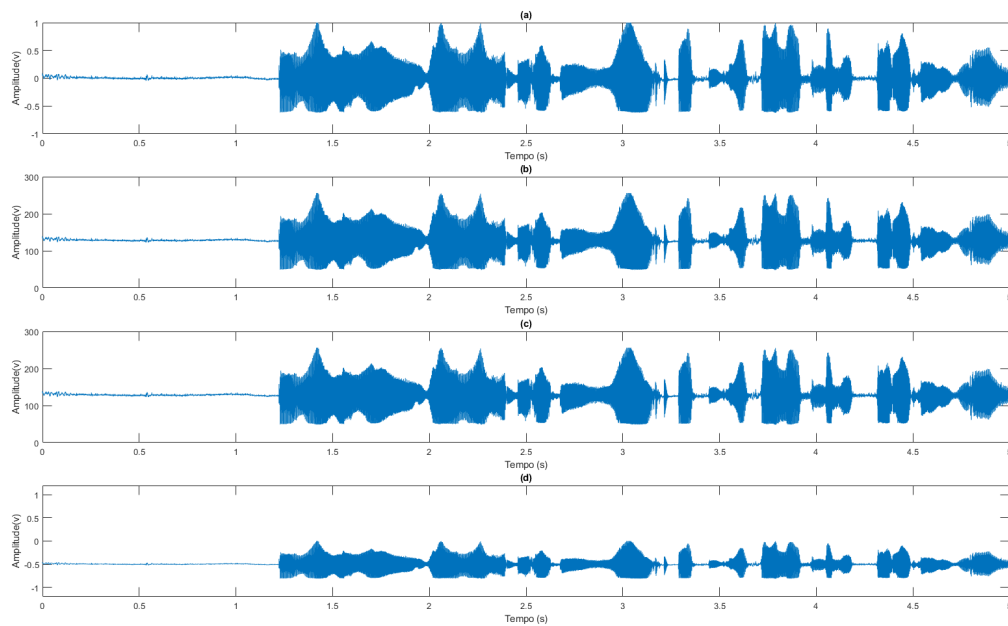


Figura 4 Setup 3 - SNR 100 e $k=8$ bits. (a) Sinal original, (b) Sinal quantizado, (c) Sinal codificado em decimal e (d) Sinal recebido.

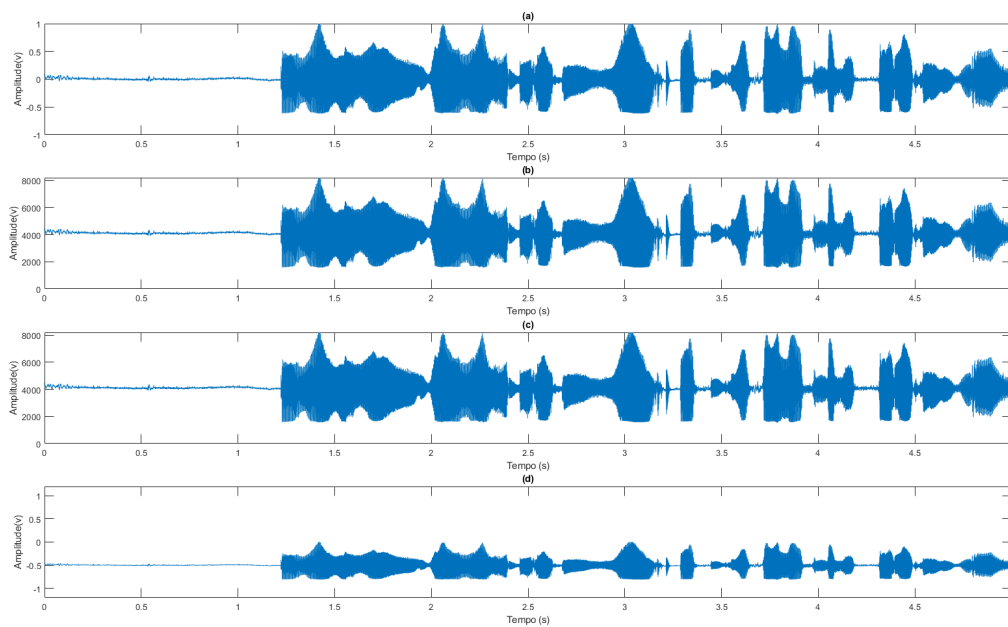


Figura 5 Setup 4 - SNR 100 e $k=13$ bits. (a) Sinal original, (b) Sinal quantizado, (c) Sinal codificado em decimal e (d) Sinal recebido.

Podemos observar que quanto mais bits adicionamos para os níveis de quantização, mais rica é a informação recebida. O sinal codificado no tempo apresenta

um formato mais pixelizado quando há menos bits. Analisando os áudios recebidos na simulação, o comportamento percebido se parece com picotes e chiados nos áudios. A figura a seguir mostra os sinais transmitidos em NRZ.

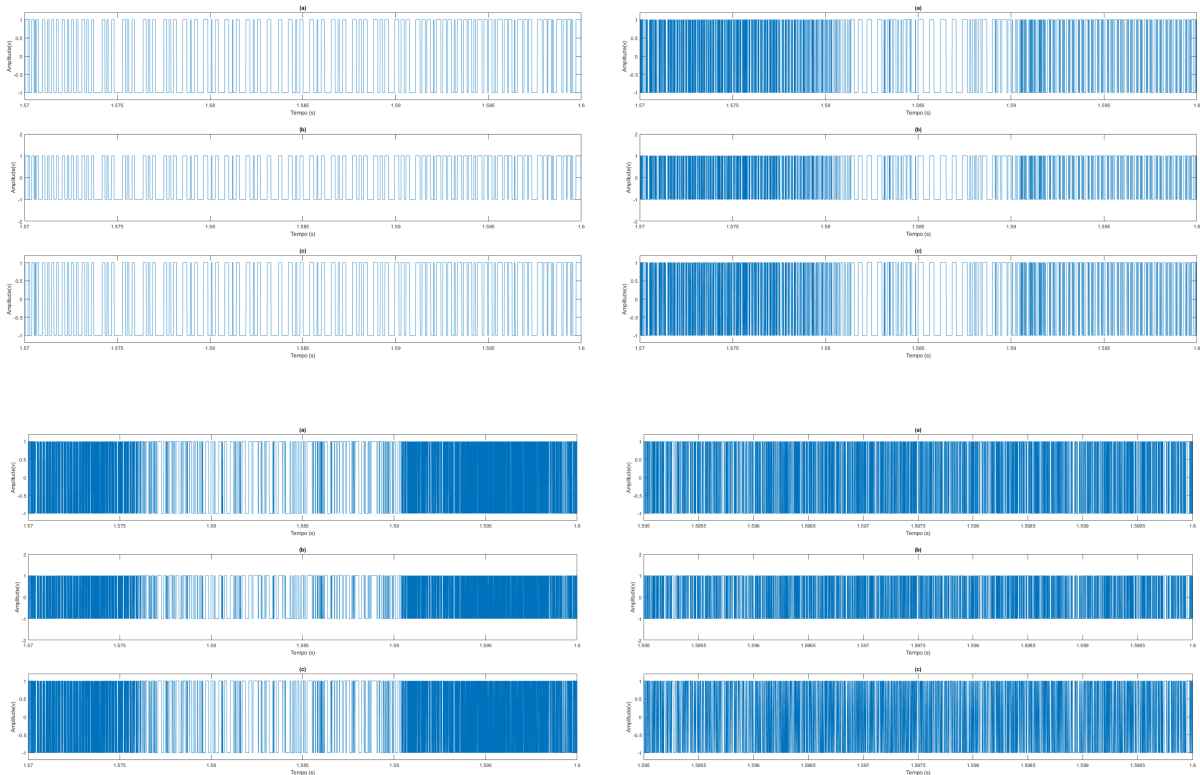


Figura 6 Sinal transmitido NRZ. Parte superior: 3 e 5 bits. Parte inferior: 8 e 13 bits.

Para a segunda parte do experimento, foi mantido 8 bits para os níveis de quantização e realizada a variação da SNR com os valores: 0, 5 e 10. As imagens a seguir mostram como ficaram os sinais realizando a variação da SNR.

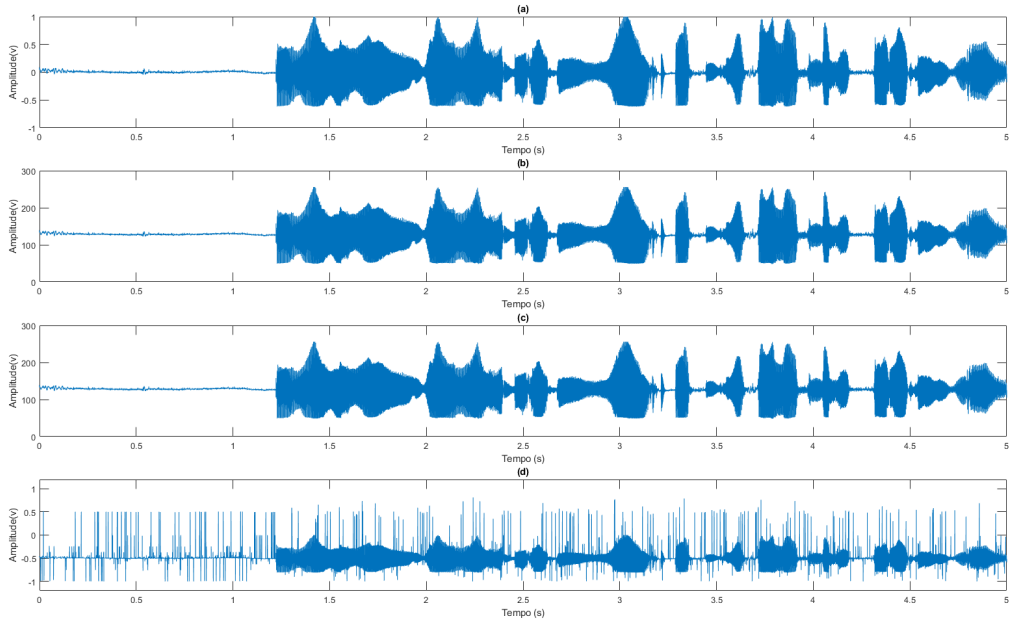


Figura 7 Setup 1 - SNR 0 e $k=8$ bits. (a) Sinal original, (b) Sinal quantizado, (c) Sinal codificado em decimal e (d) Sinal recebido.

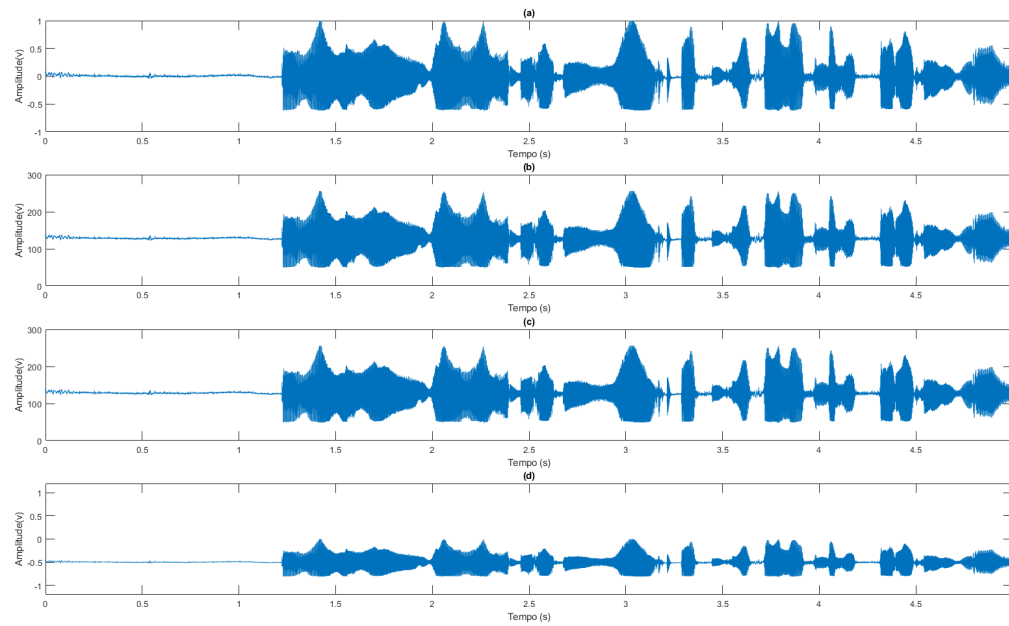


Figura 8 Setup 2 - SNR 5 e $k=8$ bits. (a) Sinal original, (b) Sinal quantizado, (c) Sinal codificado em decimal e (d) Sinal recebido.

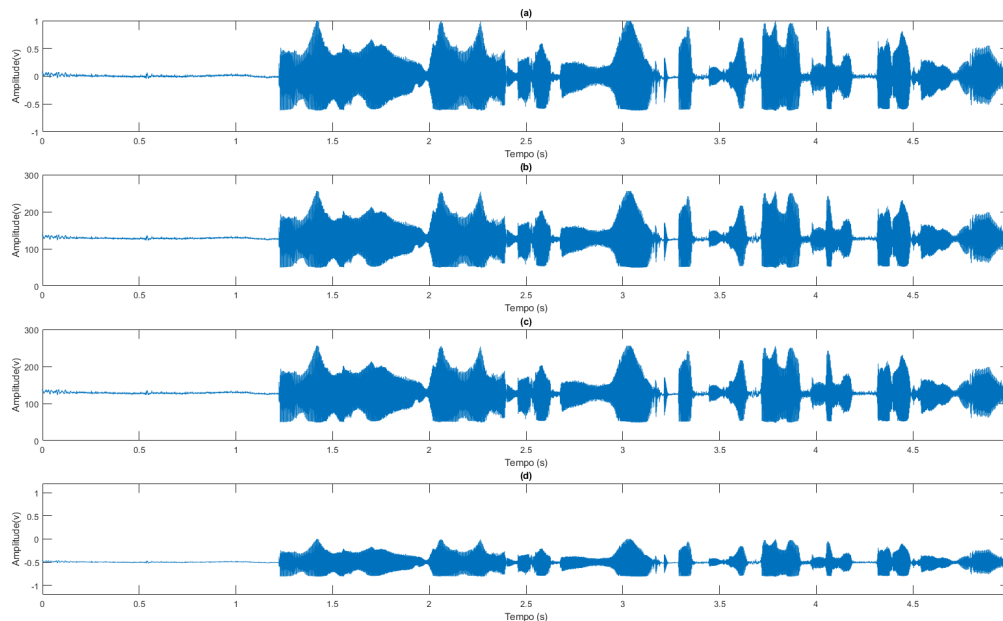


Figura 9 Setup 3 - SNR 10 e $k=8$ bits. (a) Sinal original, (b) Sinal quantizado, (c) Sinal codificado em decimal e (d) Sinal recebido.

Podemos observar que o sinal da figura 7 possui maior ruído no sinal recebido e que nas demais é visualmente imperceptível o ruído. Analisando os áudios no receptor após a decodificação, apenas o sinal com $\text{SNR}=0$ mostrou um pouco de granulação no áudio. Já os demais sinais não demonstraram nenhum tipo de alteração após a filtragem. Lembrando que isso só foi possível porque foi utilizado filtro casado no receptor, não sendo realizada a análise do áudio sem ter passado pelo filtro casado.

5. Conclusão

Este trabalho teve como objetivo realizar a simulação de uma modulação PCM, de forma a coletar um sinal de áudio, fazer o processo de amostragem, quantização e codificação, além de realizar a transmissão NRZ unipolar e a decodificação do mesmo.

No primeiro experimento o foco foi em realizar o processamento e transmissão do sinal utilizando a variação do número de bits. Foi possível verificar que a partir de 8 bits de quantização, o usuário não consegue perceber quaisquer alterações ou interferências no áudio.

No segundo experimento, foi realizada a variação da SNR mantendo um bom valor para nível de quantização. Foi verificado que apenas a $SNR=0$ mostrou deficiências no áudio recebido que puderam ser percebidas pelo usuário.

6. Referências

- [1] SOUZA, F. A. MODULAÇÃO POR CÓDIGO DE PULSO - PCM , curso de telefonia digital. IFSC.
- [2] SKLAR, B. Digital Communications: Fundamentals and Applications. 2. edition. Prentice Hall. 1029 p.ISBN 0-13-084788-7.