

# **UNIVERSIDADE FEDERAL DE ITAJUBÁ**

## **PBLE04 – PROJETO DE MODULADOR CONFIGURÁVEL EM FPGA**

### **PARTE 1 – MODULAÇÃO PCM**

Lincoln Wallace Veloso Almeida – 2018018715

Bruno de Mello Duarte – 2016010988

Ítalo Barbosa Barros – 2018008924

Gabriel Medeiros Cardoso – 2018014574

Itajubá, 13 de setembro de 2021

A primeira etapa do projeto de rádio definido por software (SDR), consistirá na modulação PCM (Pulse-Code Modulation), onde o sinal é modulado através de três etapas: Amostragem, Quantização e codificação.

O processo de amostragem, trata-se da etapa em que o sinal analógico presente em um tempo contínuo será amostrado em tempos discretos a cada intervalo fixo de tempo, intervalo esse o qual denomina-se tempo de amostragem  $t_s$ , que é calculado tomando em conta a frequência de amostragem escolhida para aquele sinal. Para que o sinal possa ser amostrado de forma a ser possível de ser recuperado posteriormente sem que suas características se percam é necessário que a frequência de amostragem escolhida obedeça ao critério de Nyquist, que diz que para que essas características sejam preservadas de forma satisfatória no momento da amostragem é necessário que a frequência de amostragem seja ao menos 2x maior que a componente de maior frequência presente no sinal. Logo temos:

$$f_s \gg 2 * f_m$$

Sabe-se que o sinal é dado por:

$$m(t) = 3 \cos(20\pi t) + 2 \cos\left(40\pi t + \frac{\pi}{6}\right)$$

Logo, conclui-se que  $f_m = 20[Hz]$

Com a finalidade de se preservar as características do sinal modulante  $m(t)$  decidiu-se realizar a amostragem a uma frequência 40x maior que  $f_m$  logo:

$$f_s = 40 * f_m = 800[Hz]$$

O que nos dá um período de amostragem de:

$$t_s = \frac{1}{f_s} = \frac{1}{800}$$

$$t_s = 0,00125 [s]$$

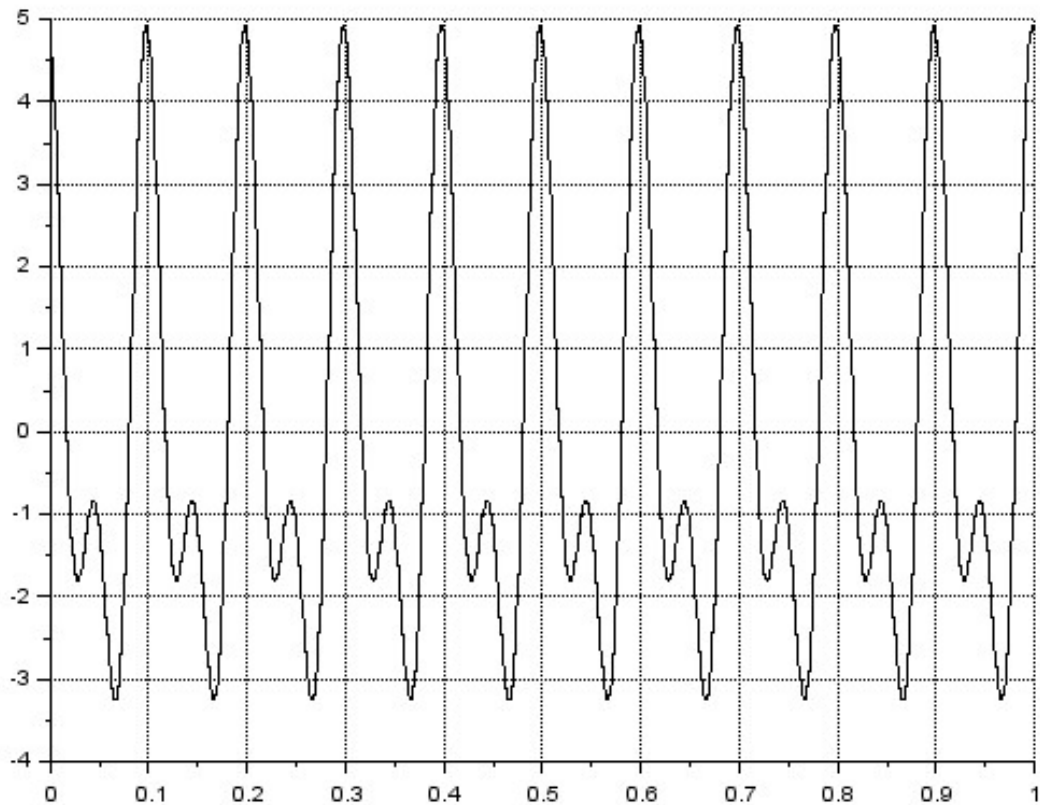
Quanto ao tempo total de amostragem, será utilizado 1[s].

Utilizando o Scinotes (script do scilab) temos:

```

fm = 20;
fs = 40*fm;
ts = 1/fs;
t = 0:ts:(1-ts);
m = 3*cos(20*%pi*t) + 2*cos(40*%pi*t + (%pi/6));
plot2d2(t,m);

```



No processo de quantização, define-se primeiramente a resolução digital, ou seja, o número de bits  $n$  que se deseja para o projeto e então tem-se o número de níveis de quantização para o sinal modulante. De acordo com as especificações do projeto serão utilizados 8 bits por amostra portanto:

$$L = 2^n = 2^8 = 256$$

O que resulta em intervalos de tensão a serem amostrados com base no valor de máximo e mínimo da tensão do sinal, ou seja:

$$\Delta V = \frac{V_{Max} - V_{min}}{L - 1}$$

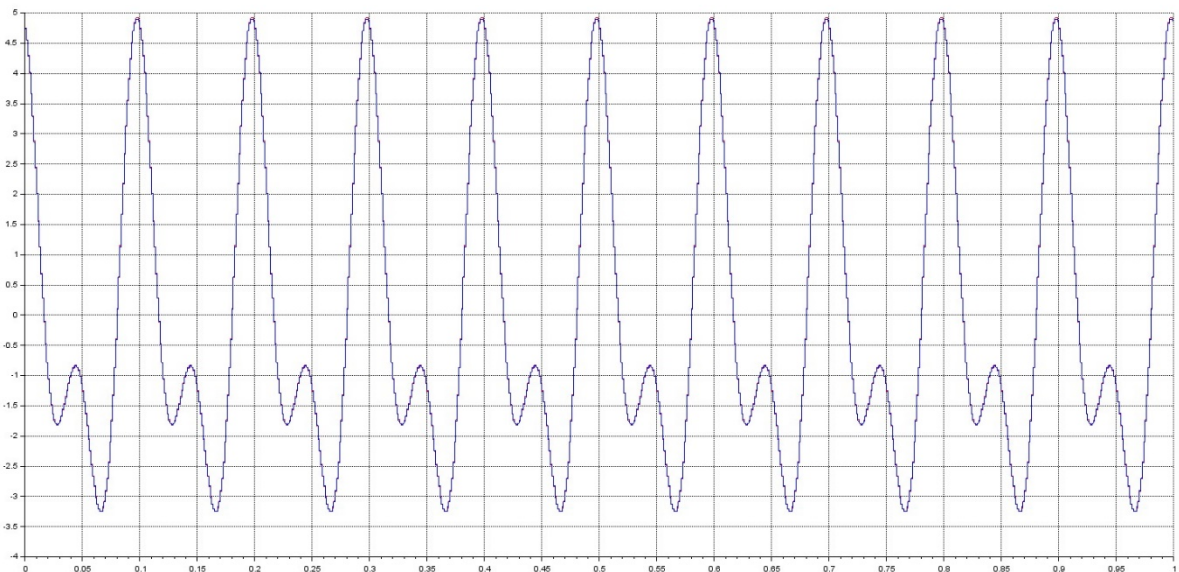
Logo, temos a implementação deste cálculo utilizando o Scilab:

```

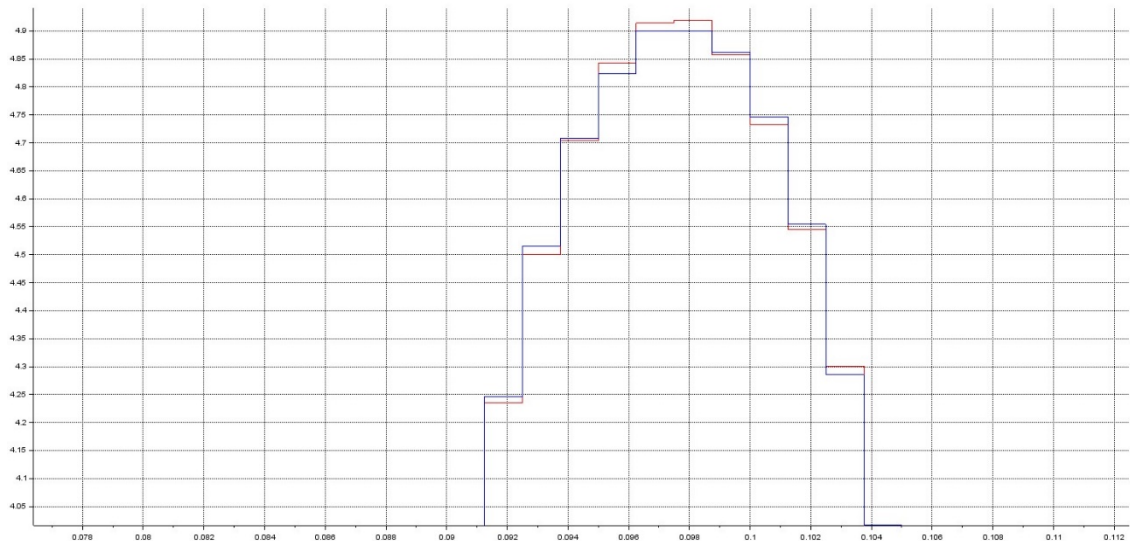
n = 8;
L = 2^n;
mmax = max(abs(m));
mq = m/mmax;
cod = mq;
d = 2/L;
q = d*[0:L-1];
q = q - ((L-1)/2)*d;
for i = 1:L
    mq(find(((q(i)-d/2)<=mq)&(mq<=(q(i)+d/2))))=
q(i).*ones(1,length(find(((q(i)-d/2)<=mq)&(mq<=(q(i)+d/2)))));
    cod(find(mq==q(i)))=(i-1).*ones(1,length(find(mq==q(i))));
end
mq = mq*mmax;
plot2d2(t,mq);

```

Obtendo-se, portanto, o sinal:



Realizando-se um zoom para verificar e comparar os sinais amostrados e quantizados temos:



Na última etapa, tem-se então a codificação dos  $n$  bits onde  $n$  é o número de bits por amostra e  $L$  o número de níveis quantizados no qual:

$$n = \log_2 L$$

Para isso, utiliza-se da implementação da rotina do Scilab, presente abaixo, onde se realiza a varredura do sinal e a cada amostra é atribuído um valor de 0x00 à 0xFF de acordo com a amplitude da amostra:

```
c = zeros(length(m),n);
for i = 1:length(m)
    for j = n:-1:0
        if(fix(cod(i)/(2^j))==1)
            c(i,(n-j)) = 1;
            cod(i) = cod(i)-2^j;
        end
    end
end
```

Logo, o valor codificado do sinal estará salvo na variável **c** que será utilizada posteriormente no projeto. Abaixo encontra-se um link onde é possível visualizar essa variável.

[variável c](#)