



Práctica 3 - Teoría de la Señal y Comunicaciones

Moisés Gautier Gómez

Ejercicios

1. Utilizado Simulink se realizará un cuantificador uniforme de 4 bits para una señal sinusoidal de amplitud 1 y frecuencia 1 Hz. Se estudiará la forma de onda de la señal error así como la SNRQ y su relación con el número de bits de cuantificación para 3 y 2 bits.

La función de entrada será la función seno:

$$f(t) = \sin(2\pi t)$$

con una amplitud 1 y una frecuencia de 1 Hz.

Δ es el tamaño del cuanto y depende del rango dinámico de la señal (máxima amplitud) y del número de bits utilizados para la codificación, por tanto:

$$\Delta = \frac{2 \cdot X_{max}}{2^B} \text{ siendo } X_{max} = 1$$

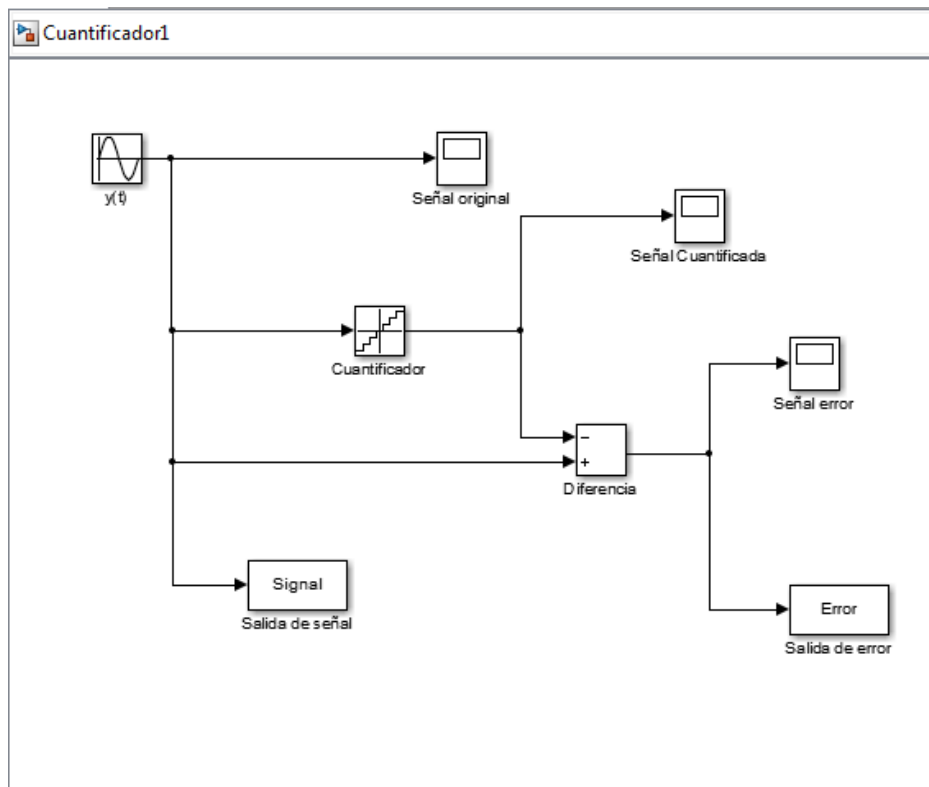
La SNRQ se define como la relación señal-ruido de cuantificación y es una medida indicativa de la bondad de la cuantificación. Se define como la razón entre la varianza de la señal de entrada y la varianza de la señal error de cuantificación:

$$SNRQ = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right)$$

En el archivo SNRQ.m se ha implementado la función correspondiente:

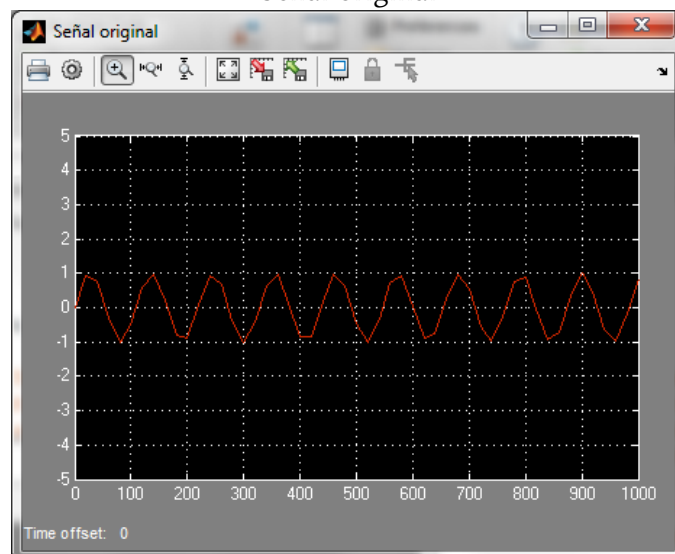
```
1 function y = SNRQ(entrada,error)
2 %La SNRQ se define como la relacion senal-ruido de ...
   cuantificacion
3 % y es una medida indicativa de la bondad de la ...
   cuantificacion.
4 % Se define como la razon entre la varianza de la ...
   senal de entrada y la
5 % varianza de la senal error de la cuantificacion
6 varianza_signal = var(entrada);
7 varianza_error = var(error);
8
9 y = 10 * log(varianza_signal/varianza_error) / log(10);
```

Cuantificador



Simulaciones:

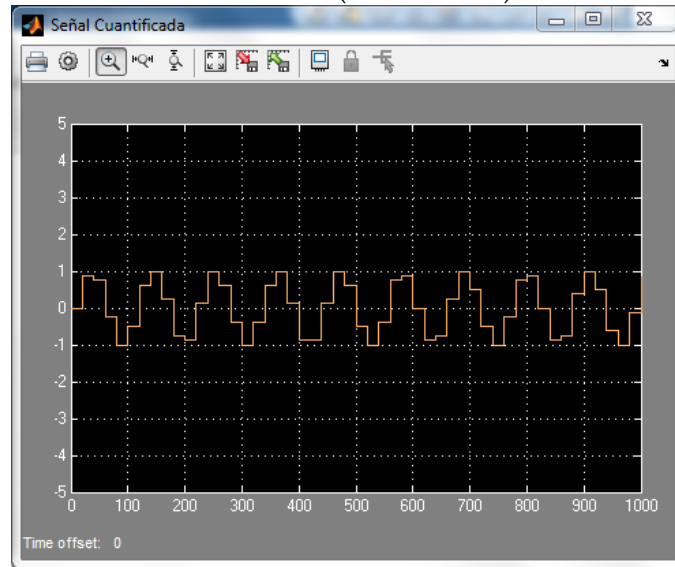
Señal original



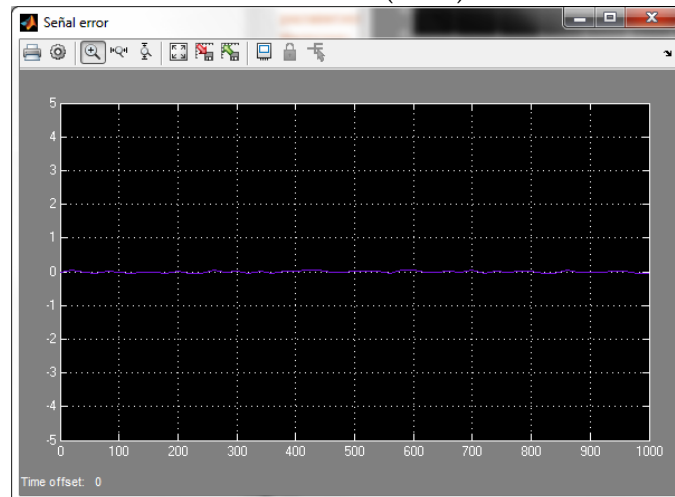
Con 4 bits

$$\Delta = \frac{2 \cdot X_{max}}{2^B} = \frac{2 \cdot 1}{2^4} = 2^{-3} = 0,125; SNRQ = 26,4493$$

Señal 4 bits (cuantificada)



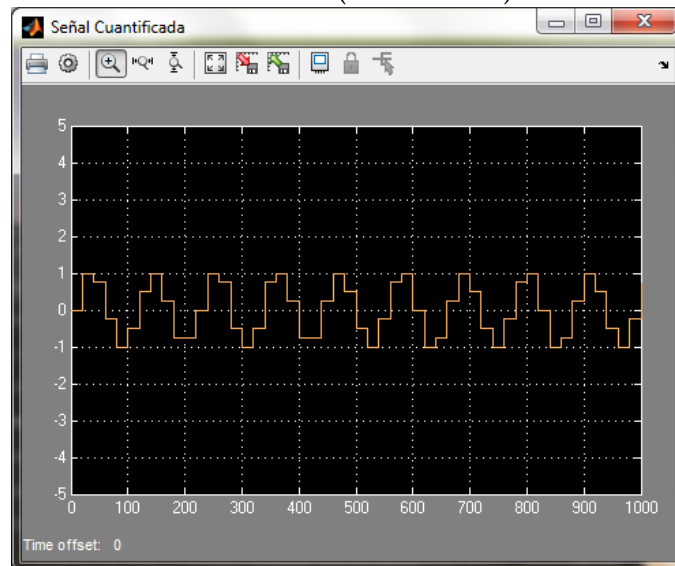
Señal 4 bits (error)



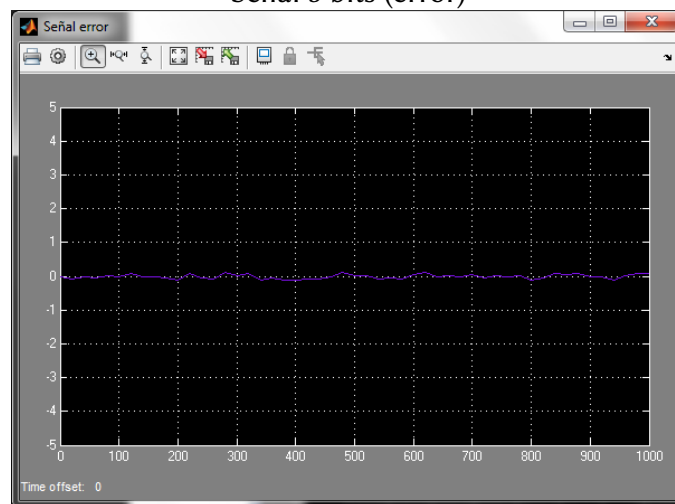
Con 3 bits

$$\Delta = \frac{2 \cdot X_{max}}{2^B} = \frac{2 \cdot 1}{2^3} = 2^{-2} = 0,25; SNRQ = 20,6633$$

Señal 3 bits (cuantificada)



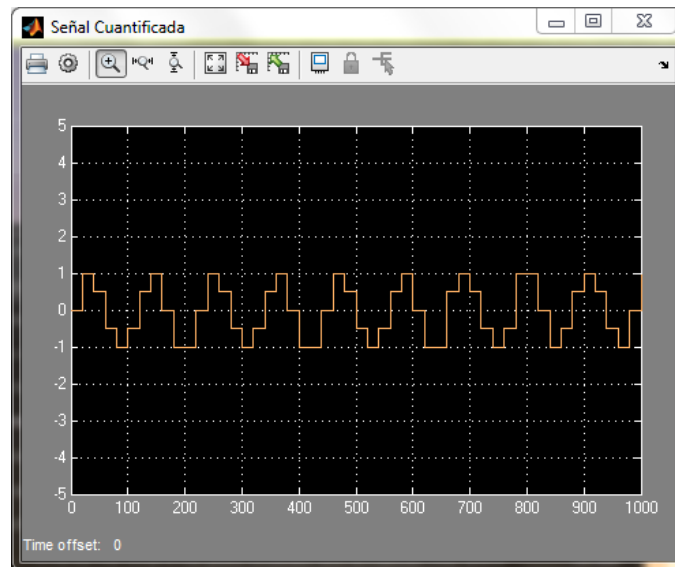
Señal 3 bits (error)



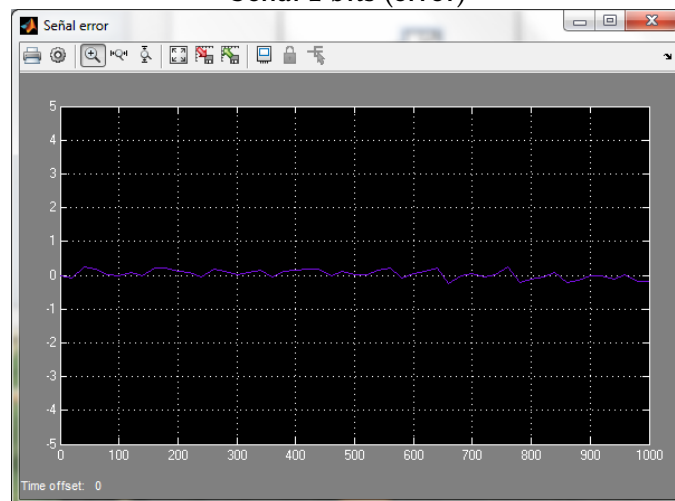
Con 2 bits

$$\Delta = \frac{2 \cdot X_{max}}{2^B} = \frac{2 \cdot 1}{2^2} = 2^{-1} = 0,5; \text{ SNRQ} = 14,8274$$

Señal 2 bits (cuantificada)



Señal 2 bits (error)



2. Utilizando el esquema de cuantificación diseñado anteriormente se estudiará el efecto de la cuantificación uniforme de una señal de voz (que se tomará del fichero `asci signal.asc`). Esta señal está cuantificada con 16 bits, se propone la recuantificación a 12, 10 y 8 bits de la siguiente manera:

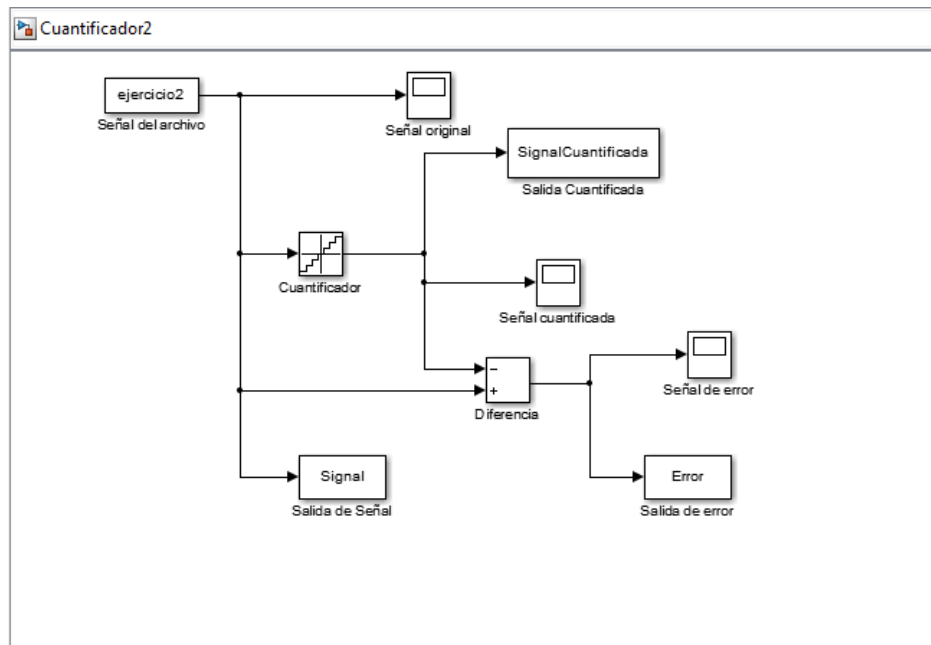
- a) 12 bits: el cuanto se fija a $2^{16-12} = 16$.
- b) 10 bits: el cuanto se fija a $2^{16-10} = 64$.
- c) 8 bits: el cuanto se fija a $2^{16-8} = 256$.

Estudie la SNRQ de cuantificación para cada uno de los casos y reproduzca la señal cuantificada en la tarjeta de audio. Comente los resultados.

Para realizar este ejercicio he realizado una función que guarda los valores de la señal proveniente del archivo a un tipo matriz[tiempo,valor] en el espacio de trabajo de Matlab.

```
1 function ejercicio2 = cargar()
2
3 % Cargamos el archivo que se nos pide en el ejercicio
4 load signal.asc;
5
6 % La frecuencia a asignar
7 fs = 8000;
8
9 % Numero de pulsos
10 n = length(signal);
11
12 % Ajustamos el tiempo en funcion de la frecuencia
13 time = 0:1/fs:(n-1)/fs;
14
15 ejercicio2 = [time',signal];
16
17 % Ahora vamos a reproducir el sonido por los ...
    altavoces de nuestro sistema
18
19 altavoz = ejercicio2./fs;
20
21 % El primer parametro de la funcion nos define el ...
    canal de audio
22 % un vector (mono) y una matriz (stereo); el segundo ...
    parametro
23 % nos define el espectro de audio en el que vamos a ...
    representarlo
24 % (audio mas lento o mas rapido dependiendo del espectro)
25 % y el ultimo parametro nos dice el numero de ...
    muestras o bits
26 % que se tomaran para hacer la representacion.
27
28 sound(altavoz,fs,8);
```

Cuantificador:



Simulaciones:

Con 12 bits.

$$\Delta = 16; SNRQ = 55,6538$$

Con 10 bits.

$$\Delta = 64; SNRQ = 44,3696$$

Con 8 bits.

$$\Delta = 256; SNRQ = 33,2319$$

En la señal original se escucha la secuencia 39920361, y al cuantificar podemos comprobar que se escucha la misma secuencia de igual forma.

- Usando simulink, diseño un cuantificador ley μ para la señal de voz (signal.asc).

Para calcular la señal de entrada, emplearemos la siguiente función:

$$F[x(n)]n = X_{max} \cdot \frac{\log[1 + \mu \frac{|x(n)|}{X_{max}}]}{\log(1 + \mu)} \cdot \text{sig}[x(n)]$$

Con $X_{max} = 2^{15} = 32768$ y $\mu = 255$. Así pues tenemos que:

$$\begin{aligned} F[x(n)]n &= 32768 \cdot \frac{\log[1 + 255 \frac{|x(n)|}{32768}]}{\log(1 + 255)} \cdot \text{sig}[x(n)] \\ &= 5909,3 \cdot \log(1 + 0,0078 \cdot |x(n)|) \text{sig}[x(n)] \end{aligned}$$

La señal cuantificada se calcula aplicando la inversa de esta función:

$$y(n) = 5909,3 \cdot \log(1 + 0,0078 \cdot |x(n)|) \cdot \text{sig}[x(n)]$$

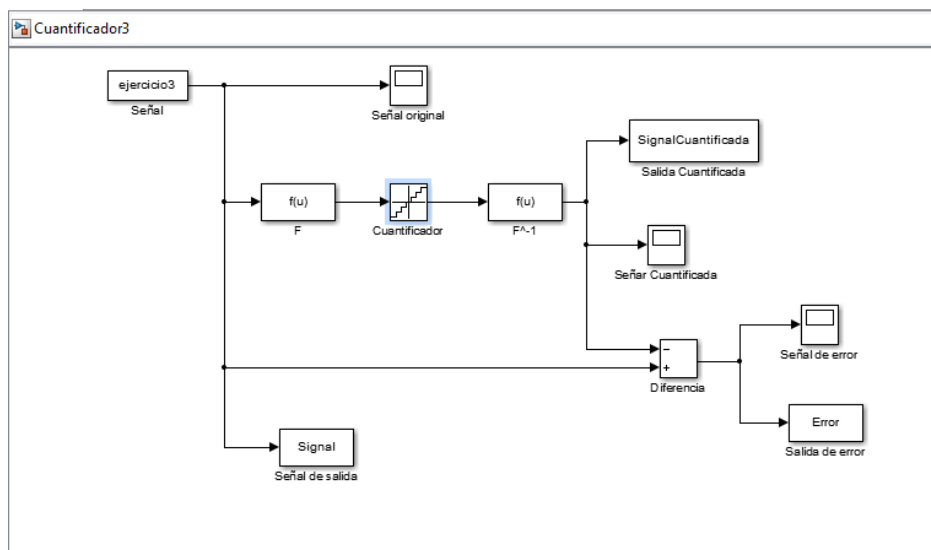
$$\frac{|y(n)|}{5909,3} = \log(1 + 0,0078 \cdot |x(n)|)$$

$$e^{\left(\frac{|y(n)|}{5909,3}\right)} = 1 + 0,0078 \cdot |x(n)|$$

$$\frac{e^{\left(\frac{|y(n)|}{5909,3}\right)} - 1}{0,0078} = |x(n)|$$

$$F^{-1}[y(n)] = \frac{e^{\left(\frac{|y(n)|}{5909,3}\right)} - 1}{0,0078} \cdot \text{sig}[y(n)]$$

Cuantificador:



4. Calcule la SNR para 12, 10 y 8 bits de cuantificación.

Como en el siguiente ejercicio tendremos que comparar los datos obtenidos aquí con los obtenidos en el ejercicio 2, utilizaré los mismos cuantos.

Simulaciones:

Con 12 bits.

$$\Delta = 16; SNRQ = 61,2672$$

Con 10 bits.

$$\Delta = 64; SNRQ = 55,1883$$

Con 8 bits.

$$\Delta = 256; SNRQ = 37,4841$$

5. Compare los resultados con los obtenidos en el apartado anterior.

Lo que nos puede llamar la atención en primera instancia es que para los valores de SNRQ se obtiene un valor mayor en cuantificación logarítmica que en cuantificación uniforme, de ahí que la aproximación sea mejor.

Así pues, observando las gráficas de error, en cuantificación uniforme el error es muy parecido para todas las amplitudes, mientras que en cuantificación logarítmica, el error es mayor cuanto mayor es la amplitud.