

Trabajo Práctico Número 1

Procesamiento Digital de Señales - Fundamentos

Prof: Pablo Slavkin

Alumno: Jacobo O. Salvador

1) Demuestre si los siguientes sistemas son LTI:

Para evaluar si es un sistema es lineal e invariante en el tiempo se de cumplir tres propiedades básicas:

Escalado

Adición

Superposición

Invariante en el tiempo

Además se debe tener en cuenta:

Fidelidad senoidal

En todo sistema LTI para una entrada senoidal la salida es siempre senoidal.

Linealidad estática

En todo sistema LTI para una entrada constante (DC) la salida es siempre la entrada multiplicada por una constante.

a)

$$y(t) = x(t) * cos(t)$$

El sistema no es LTI, cumple con propiedad de adición y escalado pero no con la invarianza si introduzco $x(t-t_0)$ la salida será $x(t-t_0)cos(t)$.

b)

$$y(t) = cos(x(t))$$

El sistema no es LTI se cumple la propiedad de invarianza en el tiempo pero no la de (escalada, adición y superposición). ES un sistemas que donde sí la señal de entrada varia en forma lineal, su salida es constante en amplitud pero variable en frecuencia, modulación por frecuencia modulada.

c)

$$y(t) = e^{x(t)}$$

No es LTI,es exponencial, no cumple con los principios de escalado, adición.

d)

$$y(t) = \frac{1}{2}x(t)$$

El sistema es LTI, cumple con los principios de escalado, adición, escalabilidad e invarianza en el tiempo.

2) Calcule la relación señal a ruido de cuantización teórica máxima de un sistema con un ADC de:

$SNR = 1,76 + 6,02 * N$

$-24 \text{ bits } SNR = 1,76 + 6,02 * 24 = 146\text{dB}$

$-16 \text{ bits } SNR = 1,76 + 6,02 * 16 = 98,56 \text{ dB}$

$-10 \text{ bits } SNR = 1,76 + 6,02 * 10 = 61,96 \text{ dB}$

$-8 \text{ bits } SNR = 1,76 + 6,02 * 8 = 49,96 \text{ dB}$

$-2 \text{ bits } SNR = 1,76 + 6,02 * 2 = 13,8 \text{ dB}$

Dado un sistema con un ADC de 10 bits, que técnica le permitiría aumentar la SNR? En que consiste?

Manteniendo la resolución dl sistema fija la única posibilidad de subir la SNR es aumentando la frecuencia de muestreo.

Calcular el filtro antialias que utilizara para su practica y/o trabajo final y justifique su decision.

En mi trabajo no utilizaré el ADC, en principio el proyecto se basa en enviar a la EDU-CIAA una señal modulada por una señal pseudoperiódica por ejemplo un tono de 200 Hz. La modulación con este tipo de señales amplia el espectro. Voy a discutir este punto en la presentación del trabajo.

Generación y simulación

– senoidal (fs[Hz], f0[Hz], amp[0 a 1], muestras),fase [radianes]

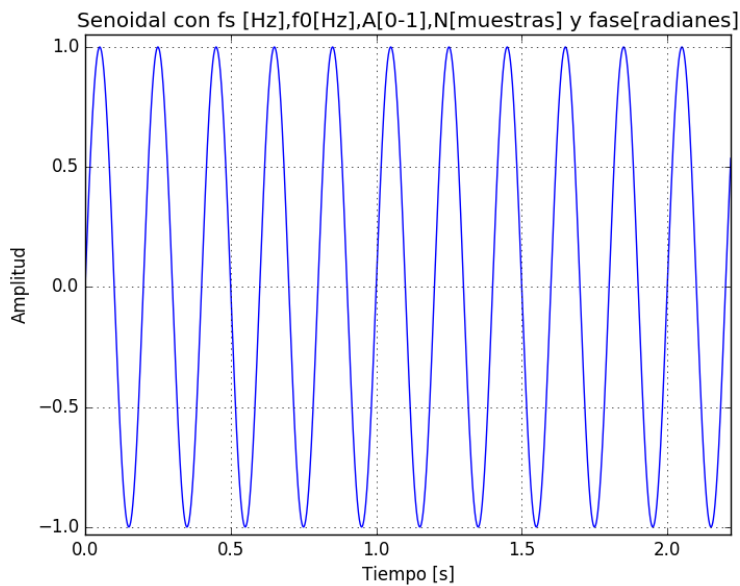
```
In [ ]: import numpy as np
import scipy.signal as sc
import matplotlib.pyplot as plt

def senoidal(fs,f0,A,N,fase):
    # fs [Hz]
    # f0 [Hz]
    # A [a-1]
    # N {muestras}
    # fase [radianes]
    t=np.arange(0,N,1/fs)
    y=A*np.sin(2 * np.pi * f0 * t + fase) #sin

    return t,y
#-----fs---f0---A---N---fase
n,y=senoidal(1000,100,1,10,0)
n1,y1=senoidal(1000,1100,1,10,0)

fig, ax = plt.subplots()
ax.plot(n,y,'b',n1,y1,'ro-')
ax.set(xlabel='Tiempo [s]', ylabel='Amplitud',title='Senoidal con fs [Hz],f0[Hz],A[0-1],N[muestras] y fase[radianes]')
ax.grid()
ax.set_ylim(-2,2)
plt.show()
```

fs=1000, f0=5, A=1, Muestras=1000, fase=0



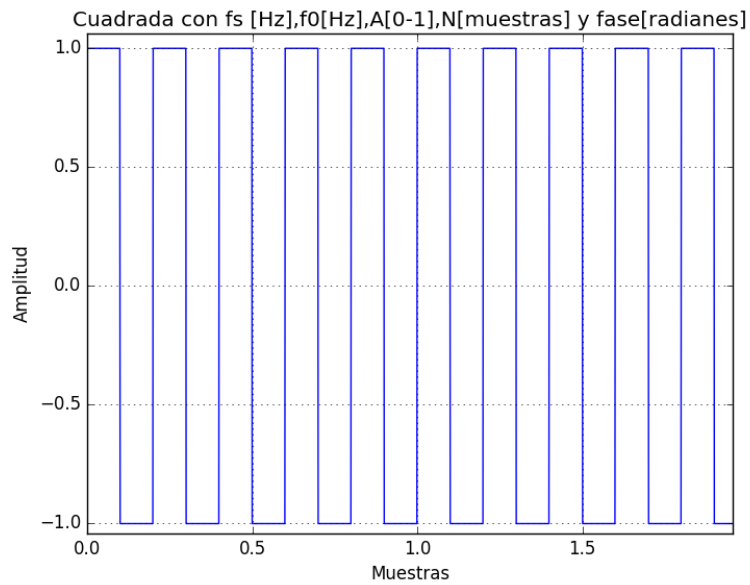
```
In [ ]: import numpy as np
import scipy.signal as sc
import matplotlib.pyplot as plt

def square(fs,f0,A,N,fase):
    # fs [Hz]
    # f0 [Hz]
    # A [a-1]
    # N {muestras}
    # fase [radianes]

    n = np.arange(0, N, 1/fs)

    y=A*sc.square(2 * np.pi * f0 * n+fase,duty=0.5)

    return n,y
#A 5 Hz waveform sampled at 500 Hz for 1 second:
#-----fs---f0---A---N---fase
n,y=square(1000,5,1,10,0)
fig, ax = plt.subplots()
ax.plot(n,y)
ax.set_ylim(-2,2)
ax.set(xlabel='Muestras', ylabel='Amplitud',title='Cuadrada con fs [Hz],f0[Hz],A[0-1],N[muestras] y fase[radianes]')
ax.grid()
plt.show()
```

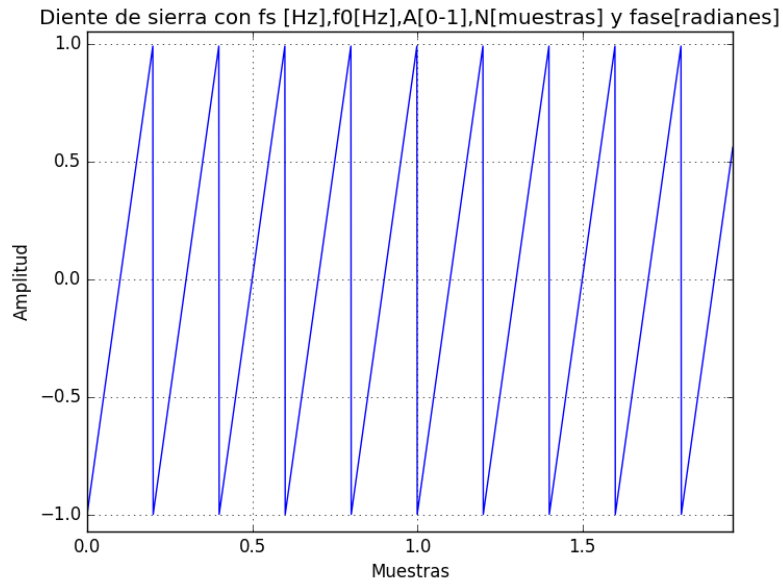


```
In [ ]: import numpy as np
import scipy.signal as sc
import matplotlib.pyplot as plt

def sawtooth(fs,f0,A,N,fase):
    # fs [Hz]
    # f0 [Hz]
    # A [a-1]
    # N {muestras}
    # fase [radianes]

    n = np.arange(0, N, 1/fs)
    y=A*sc.sawtooth(2 * np.pi * f0 * n + fase)

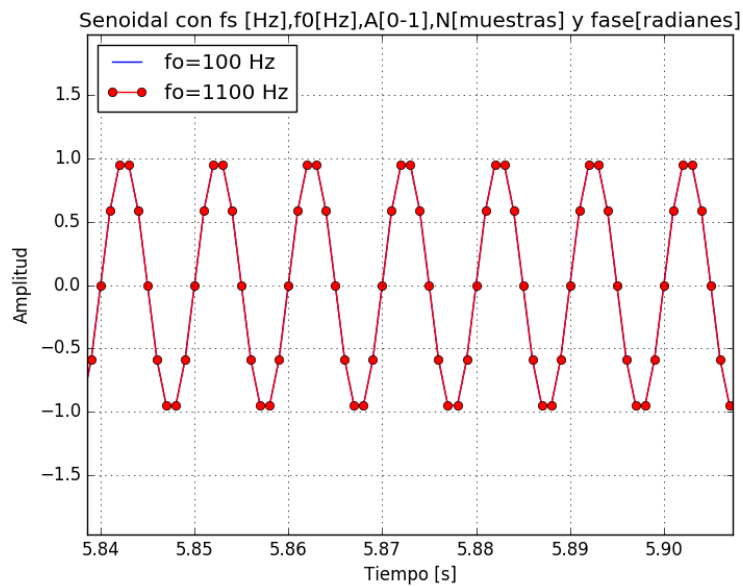
    return n,y
#A 5 Hz waveform sampled at 500 Hz for 1 second:
n,y=sawtooth(1000,5,1,10,0)
fig, ax = plt.subplots()
ax.plot(n,y)
ax.set_ylim(-2,2)
ax.set_xlabel='Muestras', ylabel='Amplitud',title='Diente de sierra con fs [Hz],f0[Hz],A[0-1],N[muestras] y fase[radianes]')
ax.grid()
plt.show()
```



Realice los siguientes experimentos

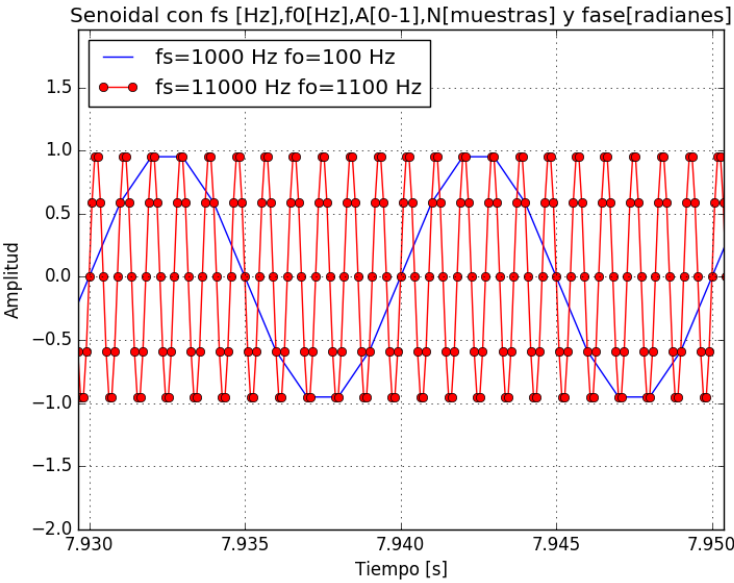
$f_0 = 0.1f_s$ y $1.1f_s$ Como podría diferenciar las senoidales?

Debajo realizamos las graficas de ambas señales.



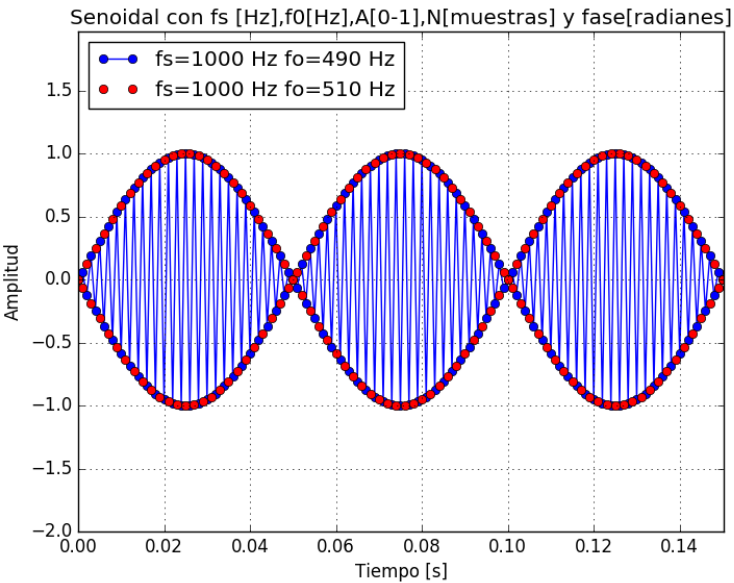
Para poder ver ambas señales correctamente debemos elegir adecuadamente el valor de fs respecto de fo. Cuando $fo=0.1fs$ tenemos una relación de 10 veces de fs respecto de fo mientras que cuando $fo=1.1fs$ solo un 10%

Hacemos la misma gráfica pero haciendo que fs en el segundo gráfico sea de fs=11000 Hz..

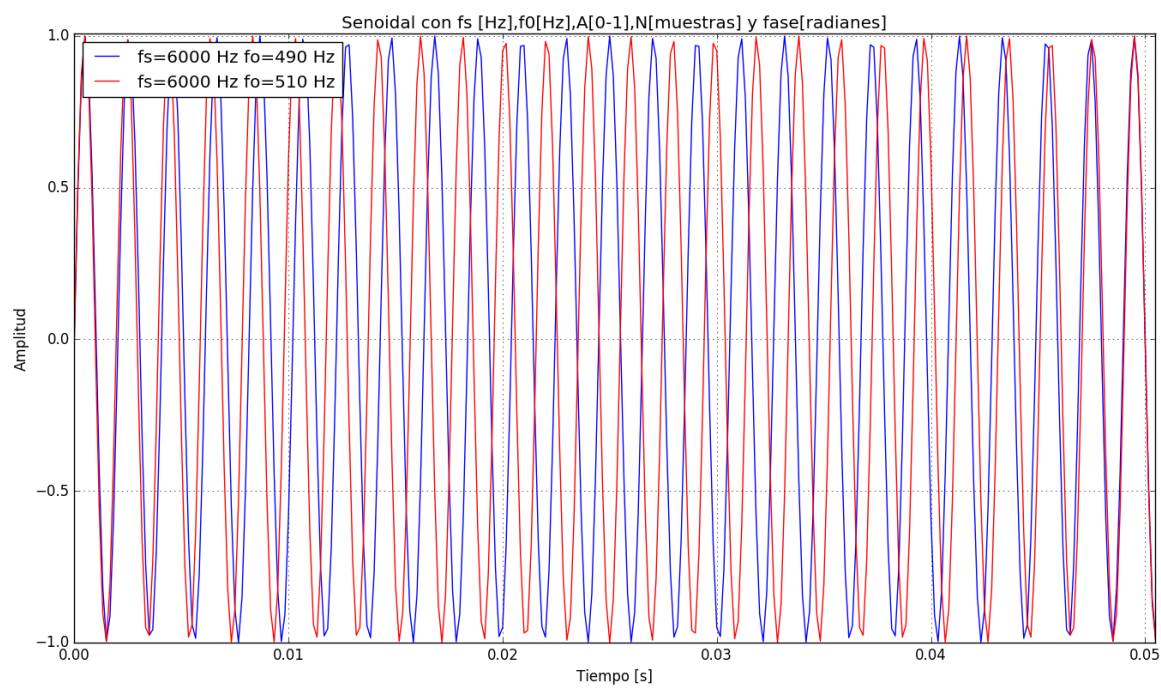


$fo = 0.49fs$ y $0.51fs$ Como es la frecuencia y la fase entre ambas?

Se toma fs= 1000 Hz



Las formas de ondas que se visualizan en la figura de arriba, no son representativas de la función sen debido a efectos que se presentan en el muestreo. Si se analiza la forma de onda en puntos color rojo se visualiza como se grafican en forma alternada. Eso se debe a que la fs o Ts muestrea los lobulos positivos y negativos en forma alternada. Los puntos de cruce por cero determinan la diferencia de fase entre ambas señales, veamos un ejemplo: Si tengo dos frecuencias $f1=490$ Hz y $f2=510$ Hz. Si hago su diferencia $f2-f1=510$ Hz - 490 Hz= 20 Hz su período es 50 ms = 0.05 s. Debajo se grafican las mismas señales pero con fs=6000.

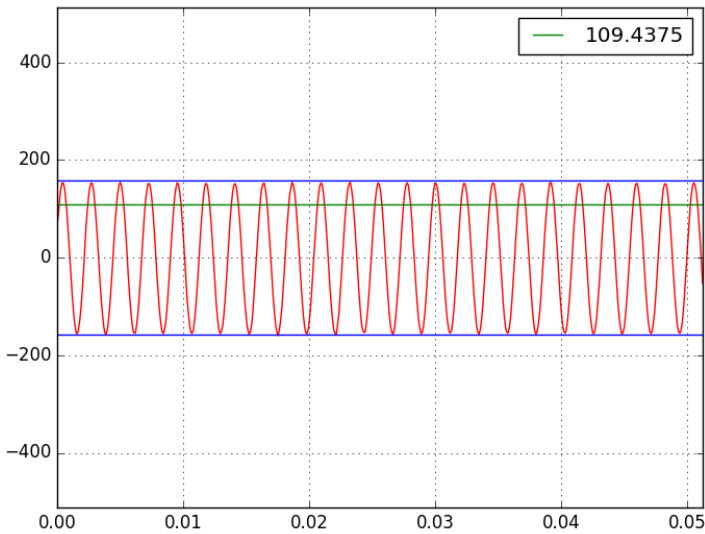


Entre el tiempo 0 y 0.05 s se puede observar como varia la fase lo cual es lógico por tener frecuencias diferentes. Si se hace un análisis por fasores, se ve como un fasor es un poco mas rápido que otro haciendo que en algún momento coincidan.

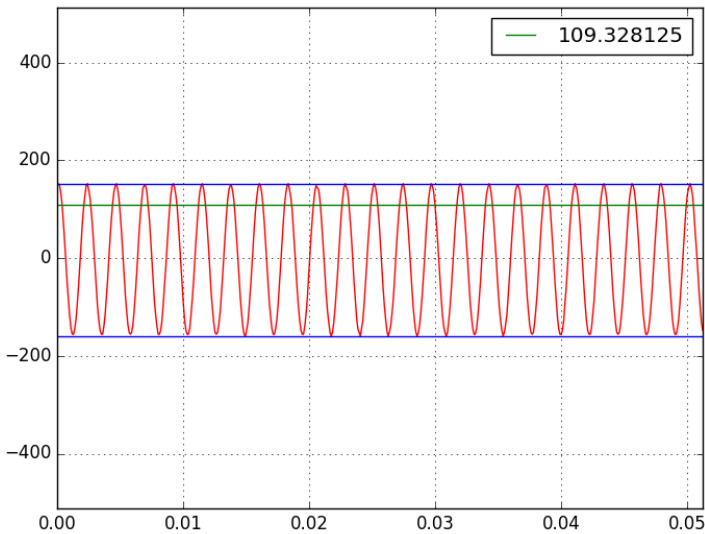
Genere con un tono de LA-440. Digitalice con 10, 8, 4 y 2 bits con el ADC, envíe los datos a la PC, grafique y comente los resultados.

En los cuatro gráficos siguientes lo que se realizó fue la gráfica de la señal de salida del ADC, enviando una señal con el simpleaudio. Se pudo verificar que por medio del ajuste del volumen se podía visualizar la saturación de la señal debido a los diodos de recorte a la entrada del circuito. En cada gráfico se cambió la resolución del ADC.

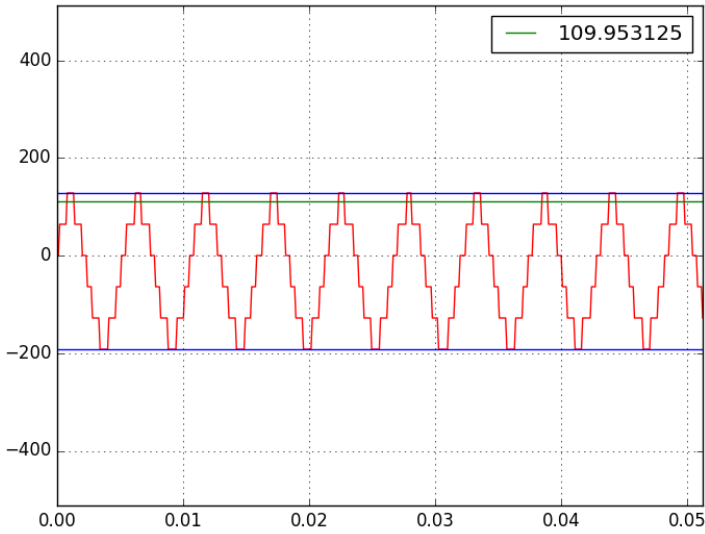
Resolución 10 bits



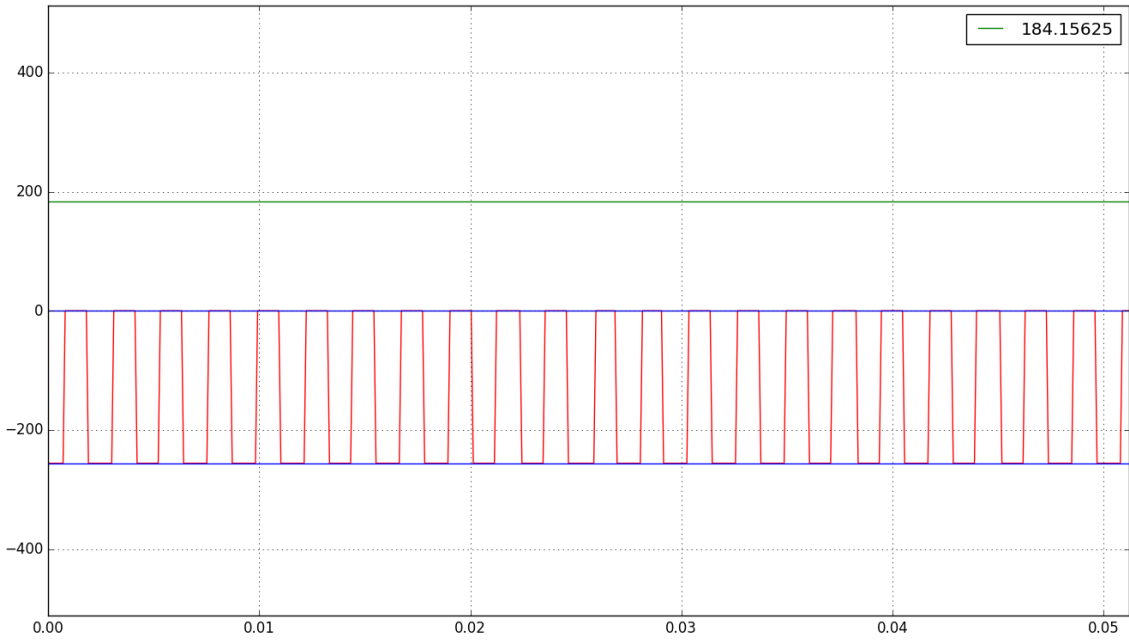
Resolución 8 bits



Resolución 4 bits

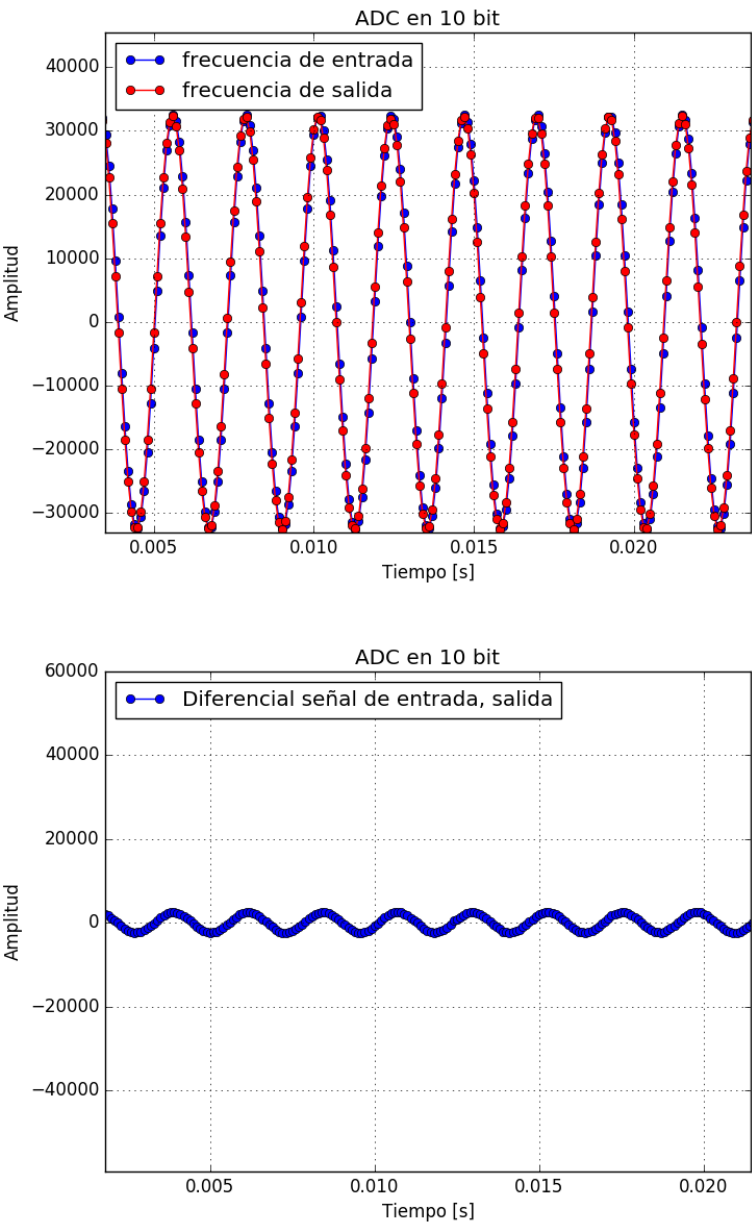


Resolución 2 bits

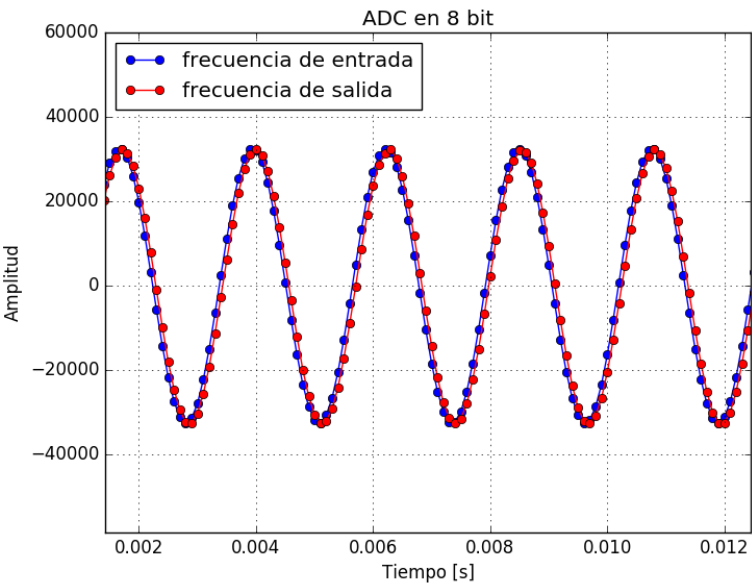


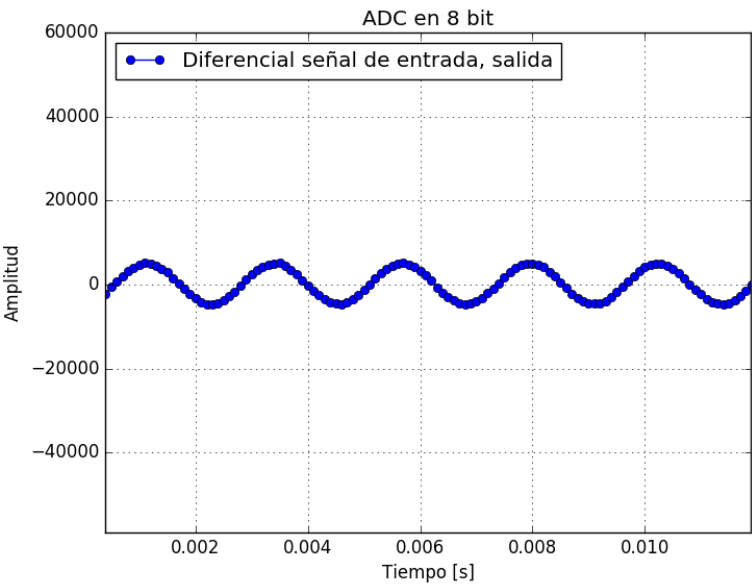
En los gráficos que siguen se gráfica la la señal de entrada en conjunto con la señal de salida. Se utilizó el circuito con lazo de salida desde el DAC a la entrada. Se gráfica la diferencia entre ambas señales.

Resolución (comparación de señales y diferencia) 10 bits

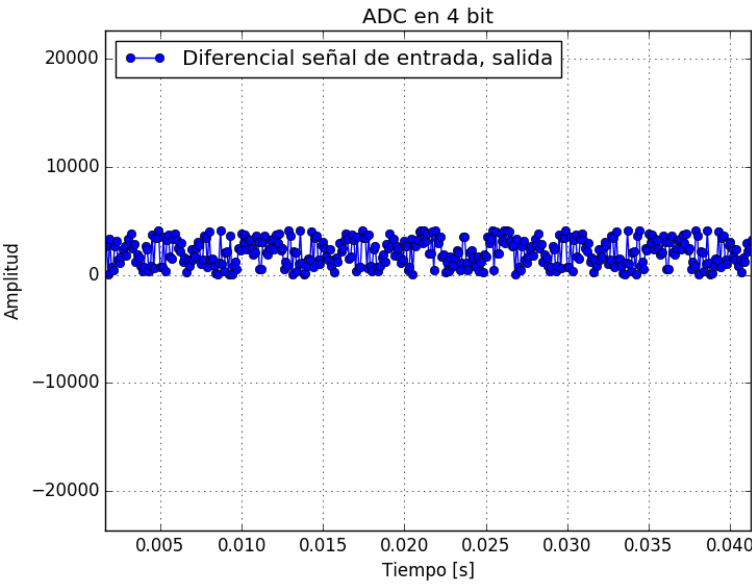
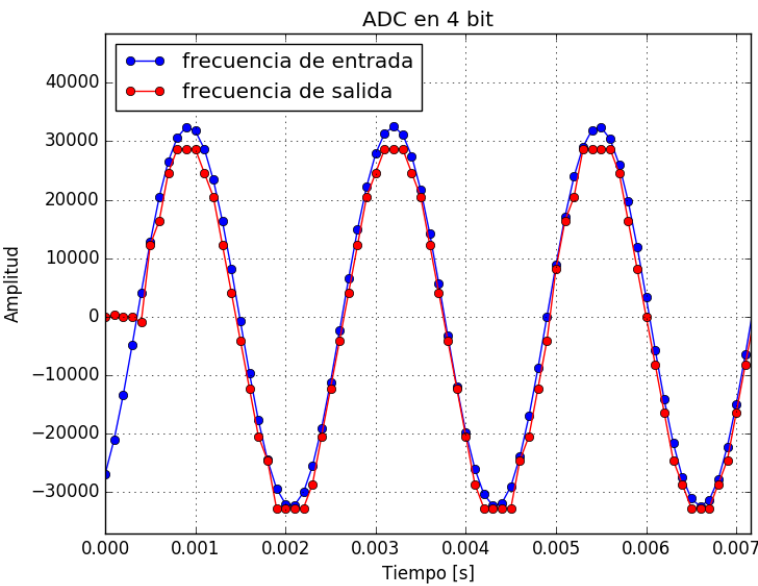


Resolución (comparación de señales y diferencia) 8 bits

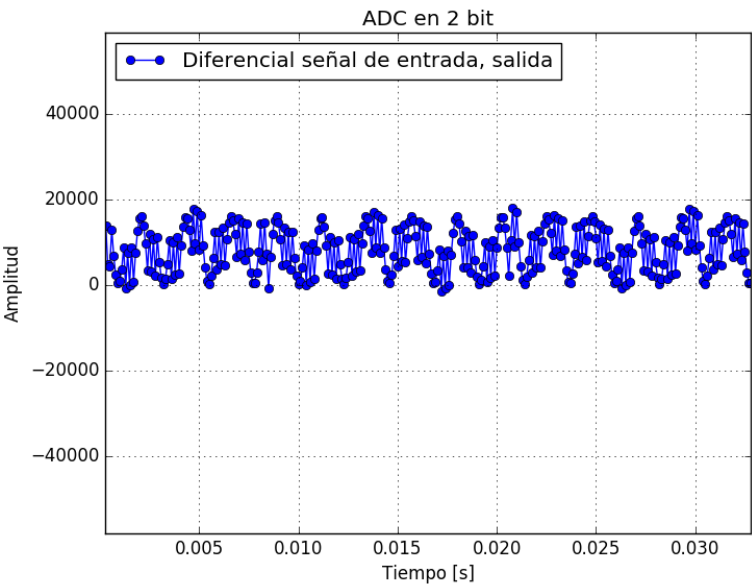
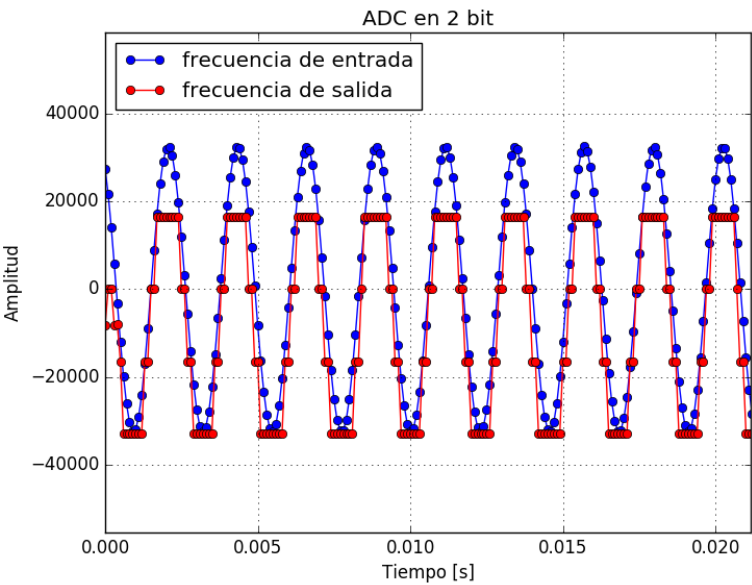




Resolución(comparación de señales y diferencia) 4 bits



Resolución (comparación de señales y diferencia) 2 bits



Explique brevemente algunas de las diferencias entre la representación flotante de simple

precision (32b) y el sistema de punto fijo Qn.m

El formato binario en coma flotante de simple precisión se utiliza por abarcar un rango de valores más amplio respecto al formato de punto fijo. En el sistema flotante de simple precisión un número es representado por su signo, un exponente y una mantisa de 24 bits. En el formato Qn.m es el utilizado generalmente en los sistemas embebidos. Si es signado, el bit de mayor orden se utiliza para el signo, y n representa la cantidad de bits para la parte entera y m representa la parte decimal.

Escriba los bits de los siguientes números decimales (o el más cercano) en float, Q1.15, Q2.14

0.5 @SQ1.15=0b0100000000000000 0.5@ SQ2.14=0b0010000000000000

-0.5 @SQ1.15=0b1100000000000000 -0.5@ SQ2.14=0b1110000000000000

-1.25 @SQ1.15 No se puede representar -1.25@ SQ2.14=0b1011000000000000

0.001 @SQ1.15=0b0000000000100000 0.001 SQ2.14=0b000000000010000 (se trunca en 15)

-2.001 si fuese UQ2.14 tendría representación. En Q1.15 y Q2.14 no se puede representar.

204000000 Este valor por ser un entero, no tiene representación por exceder el valor n.