Tarea 2 – Procesamiento Digital de Señales

Daniel Rojas Marín 2016089821

1. En Python existen diversas bibliotecas dedicadas a procesamiento de audio en general o solo leer archivos de un formato. Un ejemplo es Wave, que lee y escribe archivos WAV. Un ejemplo de como abrir y guardar un archivo con Wave es:

```
import wave

with wave.open('audio_file.wav', 'r') as audio:
    sampleRate = audio.getframerate()
    nChannels = audio.getnchannels()
    width = audio.getsampwidth()
    frames = audio.readframes(audio.getnframes())

with wave.open('new_audio_file.wav', 'w') as newAudio:
    newAudio.setframerate(sampleRate)
    newAudio.setnchannels(nChannels)
    newAudio.setsampwidth(width)
    newAudio.writeframesraw(frames)
```

2. Un método puede ser cambiar el *frame rate* (o *sampling rate*) de la señal original. Por ejemplo, si una señal tiene 20 muestras por periodo y cambiamos la frecuencia de muestreo a 40 muestras por periodo, nuestra señal original solo corresponde a la mitad del periodo total, efectivamente duplicando la frecuencia de reproducción de la señal.
3. El archivo `play440.py` responde al ejercicio, la utilización es:

   ```
   $ python play440.py [sampling frequency]
   ```

   Por ejemplo para una señal con 8kHz como frecuencia de muestreo:

   ```
   $ python play440.py 8000
   ```

4. Para el sistema:
   a. $y(n) = cos[x(n)]$

      Es **estático** pues el sistema no tiene memoria.

      Es **no lineal** ya que según el principio de superposición para una entrada $x(n) = a_1x_1(n) + a_2x_2(n)$ se tiene $y_T(n) = cos(a_1x_1(n) + a_2x_2(n))$ y la salida superpuesta $y_S(n) = a_1cos(x_1(n)) + a_2cos(x_2(n))$ las cuales son diferentes.

Es **invariante en el tiempo** puesto que la respuesta a una entrada $x_k(n-k)$ es $y_k(n) = cos(x(n-k))$ y la salida retrasada es $y(n-k) = cos(x(n-k))$ son iguales.

Es **causal** pues solo depende del valor instantáneo.

Es **estable** pues la salida está acotada a $|cos(x(n))| \leq 1$.

b. $y(n) = x(n)cos(\omega_0 n)$

Es **estático** pues no tiene memoria.

Es **lineal** pues la salida $y_T(n) = a_1x_1(n)cos(\omega_0 n) + +a_2x_2(n)cos(\omega_0 n)$ y la salida $y_S(n) = a_1x_1(n)cos(\omega_0 n) + +a_2x_2(n)cos(\omega_0 n)$ son iguales.

Es **variante en el tiempo** pues la entrada retrasada genera la salida $y_k(n) = x_k(n-k)cos(\omega_0 n)$ y la salida retrasada resulta en $y(n-k) = x(n-k)cos(\omega_0(n-k))$.

Es **causal** pues el sistema solo depende del valor instantáneo.

Es **estable** pues la entrada acotada no puede producir una salida no acotada para este sistema.

c. $y(n) = Round[x(n)]$

Es **estático** pues no tiene memoria.

Es no **lineal** ya que $y_T(n) = Round(a_1x_1(n) + a_2x_2(n))$ es diferente a $y_S(n) = a_1Round(x_1(n)) + a_2Round(x_2(n))$ tan solo con que las constantes sean no enteras.

Es **invariante en el tiempo** ya que $y_k(n) = Round(x(n-k))$ es igual a la salida retrasada $y(n-k) = Round(x(n-k))$.

Es **causal** pues solo depende del valor instantáneo.

Es **estable** pues una entrada acotada no puede generar una salida no acotada para este sistema.

d. $y(n) = x(2n)$

Es **estático** pues no tiene memoria.

Es **lineal** ya que $y_T(n) = (a_1x_1(2n) + a_2x_2(2n))$ es igual a $y_S(n) = a_1x_1(2n) + a_2x_2(2n)$.

Es **invariable en el tiempo** ya que $y_k(n) = x_1(2(n-k))$ es igual a $y(n-k) = (x_1(2(n-k))$.

Es **no causal** pues el sistema estima la salida a partir de un valor de entrada futuro $x(n+n)$.

Es **estable** pues una entrada acotada genera una salida acotada.

5. Si el sistema es invariante en el tiempo y se observa que $x_2 = 3x_3$, la salida debería seguir el principio de superposición debe resultar que $y_2 = 3y_3$, es decir $y_2 = \{3, 6, 3\}$, lo cual no sucede. Por lo tanto el sistema no es lineal.
6. El *aliasing* es un efecto que usualmente se quiere evitar, aunque tiene un par de aplicaciones donde es útil. Un ejemplo es el efecto estroboscópico: cuando un fenómeno rotativo o cíclico es observado mediante instantes con una frecuencia cercana a la frecuencia del fenómeno, se puede visualizar el objeto como estacionario o con un movimiento lento, en vez de un suavizado continuo indistinguible, efectivamente transportando las componentes de la señal desde alta frecuencia a baja frecuencia. Se ha utilizado en películas así como en animación.

   La demostración numérica es provista por Proakis-Manolakis "*Digital Signal Processing: Principles, Algorithms & Applications*" como una introducción a la forma pasabanda del teorema de muestreo:

   Se tiene una señal cuyas componentes de frecuencia están en el rango $B_1 < F < B_2$ tal que $B = B_2 - B_1$, además B (ancho de banda) cumple que $B \ll B_1 < B_2$, por lo tanto la señal es de banda angosta.

   Si la señal se muestrea a una tasa Fs superior al doble del ancho de banda (tasa de Nyquist) pero se mantiene que $Fs \ll B_1$ se va a obtener un alias del ancho de banda en el rango fundamental $0 < F < Fs/2$. Efectivamente se conserva el ancho y las componentes de frecuencia sin distorsión, pero en una rango de menor frecuencia.