

Examen 16-06-2022

Práctica

Como parte del proceso de diseño de un dispositivo de auscultación digital cardíaca, se desea incorporar el filtrado de una señal de audio proveniente del sistema de acondicionamiento del ambiente en el lugar donde se desea utilizar el estetoscopio digital.

Los requerimientos que se le imponen para el mismo son:

- Atenuar la máxima componente en la señal de ruido por debajo de 1.6uV
- Banda de interés: 10-200 Hz
- Se digitalizará en un conversor AD de 16bits y $V_{ref} = 3.3V$

Cuenta además con registros de las señales de interés, realizados a una frecuencia de muestreo de 44.1kHz para el análisis de las mismas:

- Señal de audio limpia del corazón: [heart.wav](#)
- Señal de audio del ruido de interferencia: [aa.wav](#)

Se le pide:

1. En base a los requerimientos y la información que pueda obtener de las señales de prueba*, proponga una función de transferencia de un filtro pasabanda adecuado para el sistema.
2. Separe la función de transferencia del filtro diseñado en el punto 1 en una parte pasa-bajos y otra pasa-alto. Implemente y simule la parte pasa-bajos de forma analógica. Para ello solo puede utilizar resistencias del 10% de precisión y capacitores del 20% (ver [tabla](#)). Compare la simulación con la función de transferencia deseada.
3. Se pretende re-utilizar la sección pasa-bajos realizada en el punto anterior como filtro anti-alias, utilizando una frecuencia de muestreo de 1500Hz. Analice y justifique si el filtro cumpliría con dicha función.
4. Digitalice la sección pasa-altos y genere además un segundo filtro digital de similares características, pero de tipo FIR utilizando la PyFDA. Compare la magnitud de la respuesta en frecuencia de ambos filtros, y pruebe ambas implementaciones sobre la señal de prueba [heart_1500.wav](#), graficando los espectros resultantes.

* Para cargar las señales puede utilizar los siguientes comandos:

```

archivo_audio = 'heart.wav'

fs, data = wavfile.read(archivo_audio) # frecuencia de muestreo y datos de la señal

# Definición de parámetros temporales
ts = 1 / fs # tiempo de muestreo
N = len(data) # número de muestras en el archivo de audio
t = np.linspace(0, N * ts, N) # vector de tiempo -> la función la da x espaciados
senal = data[:, 1] # se extrae un canal de la pista de audio (si el audio es
# estereo-> los estereos tienen 2 canales)
senal = senal * 3.3 / (2 ** 16 - 1) #se escala la señal a voltios (considerando un CAD
# de 16bits y Vref 3.3V)

```

Valores normalizados de Resistencias con tolerancia del 10%											
1.00	1.20	1.50	1.80	2.20	2.70	3.30	3.90	4.70	5.60	6.80	8.20
Valores normalizados de Capacitores con tolerancia del 20%											
1.00	1.50	2.20	3.30	4.70	6.80	-	-	-	-	-	-

Tabla 1 - Valores normalizados de resistencias y Capacitores

Teoría