

DSP

Funciones en python

Funciones en python

Biblioteca numpy

- **np.arange(start, stop, step)** -> crea un vector con una secuencia de números desde inicio (start) hasta fin (stop) con incremento entre valores (step)
- **np.random.normal(loc, scale, size)** -> genera un array del tamaño indicado con números aleatorios que siguen una distribución gaussiana de una cierta media (loc) y desviación estándar (scale)
- **np.linspace(inicio, fin, num)** -> genera un array unidimensional que contiene un número específico de puntos (num) equidistantes entre un rango definido (star, stop)
- **np.round()** -> se utiliza para redondear los valores en un array
- **np.clip(signal, min, max)** -> se utiliza para limitar los valores en un array dentro de un rango específico (min, max)
- **np.sin()** -> calcula el seno trigonométrico de cada elemento de un vector
- **np.var()** -> calcula la varianza de los elementos de una array

Funciones en python

Biblioteca numpy

- **awgn(signal,snr)** -> se utiliza para agregar ruido blanco gaussiano aditivo a un vector (signal) , la cantidad de ruido agregado es calculado con el SNR en dB
- **np.log10()** -> calcula el log en base 10 de los elementos de un vector
- **np.mean()** -> calcula la media (promedio) de los elementos de un vector
- **np.sqrt()** -> calcula la raíz cuadrada positiva de un número o de los elementos de una array
- **np.min()** -> Devuelve el mínimo valor de un array
- **np.max()** -> Devuelve el máximo valor de un array
- **np.digitize(x, bins)** -> se utiliza para clasificar los valores de un vector x en "bins" (intervalos o contenedores), x: Es el vector de valores a clasificar. bins: Es un vector que define los límites de los intervalos en los se van a clasificar los valores de x. Debe estar ordenado de forma creciente.

Funciones en python

Biblioteca numpy

- **np.ones()** -> se utiliza para crear un array de números con valor 1
- **np.roots()** -> se utiliza para encontrar las raíces de un polinomio, toma como entrada los coeficientes del polinomio
- **np.random.seed(0)** -> se utiliza para fijar un valor inicial en el generador de números aleatorios
- **np.random.randn()** -> genera una secuencia de números aleatorios que siguen una distribución gaussiana de media 0 y desviación 1
- **np.convolve()** -> se utiliza para calcular la convolución de dos arrays
- **np.fft.fftfreq(n,d)** -> se utiliza para calcular las frecuencias correspondientes a los componentes de la transformada rápida de Fourier (FFT) de una señal, n número de puntos en la FFT, d intervalo de muestreo
- **np.fft.fft(a,n,axis)** -> se utiliza para calcular la Transformada Rápida de Fourier (FFT) de una señal, a: vector de entrada, n: Número de puntos en la FFT, axis: Eje a lo largo del cual se calcula la FFT

Funciones en python

Biblioteca SciPy

- **lfilter()** -> se utiliza para aplicar un filtro digital a una señal en el dominio del tiempo
- **Signal.freqz(n, d, worN, fs)** -> se utiliza para calcular y graficar la respuesta en frecuencia de un filtro digital, n: coeficientes del numerador del filtro, d coeficientes del denominador del filtro, worN: Número de puntos en los que se calculará la respuesta en frecuencia, fs: Frecuencia de muestreo
- **scipy.io.loadmat()** -> se utiliza en Python para cargar archivos de datos en formato MATLAB (.mat)
- **data['signal'][:, 0]** -> se utiliza para extraer un vector (canal) específico de un conjunto de datos cargados desde un archivo .mat
- **np.isnan(signal).any()** -> se utiliza para verificar si hay valores "NaN" (Not a Number) en un array o señal

Funciones en python

Biblioteca SciPy

- **signal.firwin()** -> se utiliza para diseñar filtros FIR utilizando la ventana de Hamming
- **signal.ellip()** -> se utiliza para diseñar filtros elípticos, que son un tipo de filtro de respuesta en frecuencia que permite obtener una atenuación especificada en la banda de paso y una cantidad de ripple (ondulación) en la banda de paso y la banda de parada
- **signal.tf2sos(b, a)** -> se utiliza para convertir un filtro en forma de función de transferencia (TF) representado por los coeficientes b (numerador) y a (denominador) a una forma en secciones de segundo orden (SOS)
- **signal.sosfilt(sos, signal)** -> se utiliza para aplicar un filtro a una señal utilizando la representación en secciones de segundo orden (SOS)

Funciones en python

Biblioteca Matplotlib

- **plt.figure()** -> Esta función crea una nueva figura para gráficos, crea un objeto de figura
- **plt.legend()** -> se utiliza para mostrar una leyenda en un gráfico
- **plt.grid(True)** -> habilitar la visualización de una cuadrícula en el gráfico
- **plt.title()** -> se utiliza para establecer el título de un gráfico
- **plt.xlabel()** -> se utiliza para establecer la etiqueta del eje x en un gráfico
- **plt.ylabel()** -> se utiliza para establecer la etiqueta del eje y en un gráfico
- **plt.xlim(min, max)** -> se utiliza para establecer los límites del eje x en un gráfico
- **plt.ylim(min, max)** -> se utiliza para establecer los límites del eje y en un gráfico
- **plt.tight_layout()** -> se utiliza para ajustar automáticamente los márgenes y el espaciado entre los subgráficos en una figura

Funciones en python

Biblioteca Matplotlib

- **plt.plot(x, y, 'linea', label='texto')** -> se utiliza para crear gráficos de líneas, línea especifica el estilo de la línea
- **plt.subplot(x,y,index)** -> se utiliza para crear múltiples gráficos en una sola figura, x: filas, y: columnas, index: índice del subgráfico
- **plt.show()** -> se utiliza para mostrar todas las figuras y gráficos que has creado hasta ese momento
- **plt.savefig("archivo.png")** -> guarda la figura actual que se ha generado en un archivo con el nombre especificado en el argumento "archivo.png"
- **patches. Circle((x,y), radius, fill)** - > se usa para crear círculos en una figura para visualización, los parámetros: (x,y): Coordenada al centro, radius, fill: relleno

Funciones en python

Biblioteca scikit-learn

- **mean_squared_error()** -> calcula el error cuadrático medio (MSE) entre dos conjuntos de datos

Biblioteca math

- **math.sqrt()** -> se utiliza para calcular la raíz cuadrada de un número

Biblioteca sounddevice

- **sd.play(signal, Fs)** -> se utiliza para reproducir audio, signal: lista que contiene los datos de la señal de audio, Fs: La frecuencia de muestreo en Hz
- **sd.wait()** -> se utiliza para esperar a que se complete la reproducción de una señal de audio