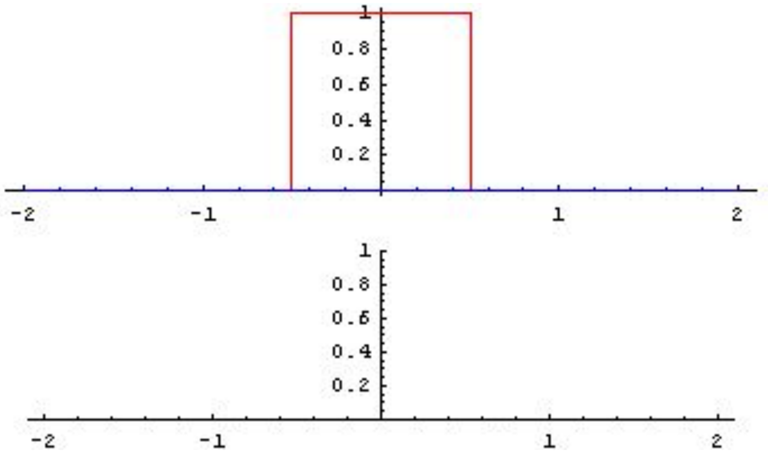
Implementación de sistemas

Señales y Sistemas
Primer semestre
2023

Ejercicio de implementación



Convolución entre dos señales y señal resultantes.

Imagen extraida de wikipedia

Simulación de sistemas en computadora

Repaso teórico: ecuación en diferencias

Sistemas IIR y FIR

Sistemas definidos mediante una ecuación en diferencias

- El ejemplo anterior es una caso particular de una clase de sistemas definido a partir de la ecuación en diferencias.
- La relación entre la entrada y la salida satisface una ecuación en diferencias de orden N de la forma

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m].$$

- Asumiendo que $a_0 = 1$ (no se pierde generalidad) se tiene que

$$\begin{aligned} y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_M x[n-M] \\ - a_1 y[n-1] - a_2 y[n-2] - \cdots - a_N y[n-N]. \end{aligned}$$

- La muestra actual de la salida se calcula como una combinación lineal de la muestra actual y M muestras previas de la entrada y N muestras previas de la salida.

Repaso teórico: ecuación en diferencias

Sistemas IIR y FIR

Sistemas definidos mediante una ecuación en diferencias

$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + \cdots + b_Mx[n-M] \\ - a_1y[n-1] - a_2y[n-2] - \cdots - a_Ny[n-N].$$

- Las constantes b_i, $i = 1, \ldots, M$ y a_j, $j = 1, \ldots, N$ son los coeficientes del sistema. El filtro queda completamente especificado con los valores de todos los coeficientes.
- Los valores b_i se llaman coeficientes de prealimentación (feedforward) y los valores a_j se llaman coeficientes de realimentación (backward).
- El filtro es recursivo si tiene algún coeficiente de realimentación no nulo. En ese caso, es un filtro IIR.
- Si todos los coeficientes de realimentación son nulos, no hay realimentación y el filtro es FIR..

Repaso teórico: ecuación en diferencias

Sistemas IIR y FIR

Sistemas definidos mediante una ecuación en diferencias

- Observaciones:
  - Los sistemas FIR siempre pueden representarse mediante una ecuación en diferencias.
    - Los coeficientes de prealimentación coinciden con la respuesta al impulso del sistema,
    $$b_k = h[k].$$
    - Los coeficientes de realimentación son nulos.

$$\begin{aligned}
y[n] &= x[n] * h[n] \\
&= \sum_{i=0}^{M} h[k]x[n-k] \\
&= h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \cdots + h[M]x[n-M] \\
&= b_0x[n] + b_1x[n-1] + b_2x[n-2] + \cdots + b_Mx[n-M]
\end{aligned}$$

Implementación de sistemas con lfilter

- Filtro causal (entradas pasadas) similar a los filtros electrónicos.
- Pueden ser de fase lineal (FIR simétrico), aunque generalmente no lo son.
- Por lo general agrega retardos a diferentes frecuencias.

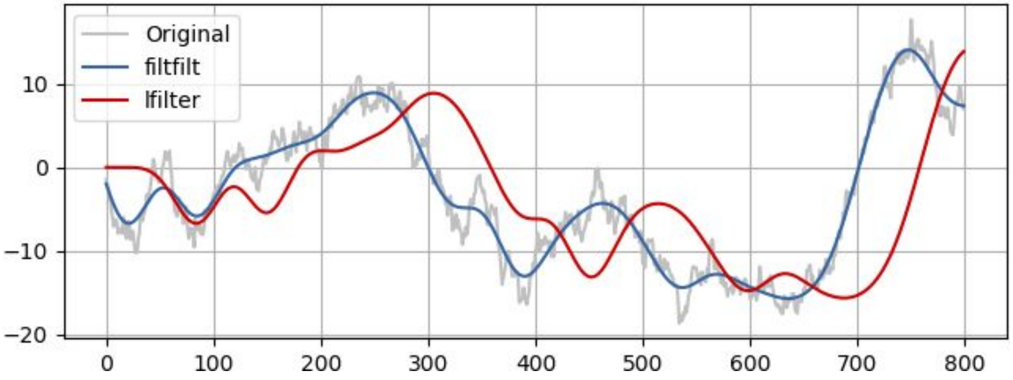


Figura 1. Señal orginal con ruido (gris), señal filtrada con filtro sin retardo (azul) y señal filtrada con lfilter (roja).

Implementación en Python

- Utilizamos el módulo lfilter de **scipy.signal** :

```
y = scipy.signal.lfilter(b, a, x)
```

donde:

- donde a y b son arrays con los coeficientes de la función de transferencia, denominador y numerador respectivamente.
- x es la señal entrada del sistema
- y es la señal de salida del sistema.

$$a_M y[n-N] + \cdots + a_2 y[n-2] + a_1 y[n-1] + \underline{a_0 y[n]} = \underline{b_0 x[n]} + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_N x[n-M]$$

$$a = [a_0, a_1, a_2, \cdots, a_N]$$ Coeficientes de las salidas

$$b = [b_0, b_1, b_2, \cdots, b_M]$$ Coeficientes de las entradas

$$Y(z) = \frac{b[0] + b[1]z^{-1} + \ldots + b[M]\,z^{-M}}{a[0] + a[1]z^{-1} + \ldots + a[N]\,z^{-N}} X(z)$$

Función de transferencia

¿Cuáles son los vectores ***a*** y ***b*** del sistema que calcula la salida copiando el valor actual de la entrada?

$$a_M y[n-N] + \cdots + a_2 y[n-2] + a_1 y[n-1] + a_0 y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_N x[n-M]$$

$a = [a_0, a_1, a_2, \cdots, a_N]$ Coeficientes de las salidas

$b = [b_0, b_1, b_2, \cdots, b_M]$ Coeficientes de las entradas

Algunos sistemas de interés

$$a_M y[n-N] + \cdots + a_2 y[n-2] + a_1 y[n-1] + a_0 y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_N x[n-M]$$

1. Identidad

$$y[n] = x[n] \longrightarrow a = [1], b = [1]$$

2. Proporcional

$$y[n] = k.x[n]$$

3. Retardo

$$y[n] = x[n-k], \quad k \in N$$

$$a = [a_0, a_1, a_2, \cdots, a_N]$$ Coeficientes de las salidas

$$b = [b_0, b_1, b_2, \cdots, b_M]$$ Coeficientes de las entradas

Ejercicios: Parte 1

1. Crear un vector de tiempo de 5s, con paso 0.01s.
2. Usando el vector anterior, crear una señal sinusoidal de frecuencia 1Hz. Desde ahora, esta señal será la entrada de cada uno de los sistemas.
3. Probar el ejemplo de la identidad en su máquina
   a. Corroborar que los resultados sean los esperados.
4. Implementar los siguientes sistemas
   a. Proporcional (k = 2)
   b. Retardo (k = 50)

4. Media móvil

$$y[n] = \frac{1}{L} \sum_{i=0}^{L-1} x[n-i], \quad L \in N$$

¿Qué coeficientes necesito?

$$a_M y[n-N] + \cdots + a_2 y[n-2] + a_1 y[n-1] + a_0 y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_N x[n-M]$$

4. Media móvil

$$y[n] = \frac{1}{L}\sum_{i=0}^{L-1} x[n-i], \quad L \in N$$

¿Qué coeficientes necesito?

$$a_M y[n-N] + \cdots + a_2 y[n-2] + a_1 y[n-1] + a_0 y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_N x[n-M]$$

$$a = [L]$$

$$b = [1, 1, \cdots, b_{L-1} = 1]$$

Ejercicios: Parte 1

1. Crear un vector de tiempo de 5s, con paso 0.1s.
2. Usando el vector anterior, crear una señal sinusoidal de frecuencia 1Hz. Desde ahora, esta señal será la entrada de cada uno de los sistemas.
3. Probar el ejemplo de la identidad en su máquina
    a. Corroborar que los resultados sean los esperados.
4. Implementar los siguientes sistemas
    a. Proporcional(k = 2)
    b. Retardo (k = 50)
    c. Media móvil (con y sin ruido)

Ejercicios: Parte 2

1. Generar una señal sinusoidal
2. Filtrarla con la identidad
3. Agregar un delay al filtrado de identidad
4. A la sinusoide de la parte 1 sumarle un ruido gaussiano de potencia 1
5. Filtrar la señal de 4.

Derivador-Acumulador

Acumulador

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

Derivador

$$y[n] = x[n] - x[n-1]$$

$$a_M y[n-N] + \cdots + a_2 y[n-2] + a_1 y[n-1] + a_0 y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_N x[n-M]$$

Ejercicios: Parte 3

1. Con un vector de tiempo de largo 20 y paso 1, construir una señal que sea una función lineal, de pendiente 1 y cruce con el eje y por cero.
2. Derivar e integrar esta señal de tres maneras:
   a. Creando dos listas vacías y calculando cada valor a mano, en un for.
   b. Usando lfilter.
   c. Con funciones de Python: investigar numpy diff y numpy cumsum.