--

CONVOLUCIÓN

--

La convolución es una operación que nos permite conocer la respuesta y[n] de un sistema lineal e invariante en el tiempo a una señal de entrada arbitraria x[n], a partir del conocimiento de la respuesta del mismo sistema a una señal de entrada básica, el impulso:

```
    x[n] -> sistema LIT -> y[n]
    donde
        y[n] = T{x[n]}
```

La función h[n] es el resultado de usar impulso como función de entrada:

```
        h[n] = T{delta[n]}
```

A partir de h[n] podemos calcular la respuesta a cualquier señal x[n] usando la operación de convolución

```
        y[n] = x[n]*h[n]
```

Definición matemática de la operación de convolución:

```
    x[n]*h[n] = sumatorio(desde k=-infinito; hasta infinito; x[k]h[n-k])
```

Datos

Vamos a particularizar a señales concretas para hacerlo más sencillo:

```
    x[n] = [_1_, 2, 3, 4, 4, 3]
    h[n] = [1, _-1_, 1, 2]
```

```
        |                       |
   x[n] |                  h[n] |
        |     * *               |
        |   * | | *             |
        | * | | | |             |   *
        * | | | | |           * | * |
     ---+-+-+-+-+-+---->     -+-+-+-+--->
        |                       *
```

Fíjate que la señal x[n] comienza en n=0 y la señal h[n] en n=-1.

Comentarios

Por limitaciones con el formato de texto, estoy indicando dónde empieza cada señal usando _subrayados_.

Método 1 ("método gráfico")

En este método para calcular la convolución x[n]*h[n] vamos a hacer uso de las propiedades de los sistemas LIT para reducir las operaciones que se deben realizar:

1. El sistema es lineal

Eso significa que, si la señal es una combinación lineal, la salida será la suma escalada de los términos de la combinación lineal. Es decir,

```
    para
        x[n] = a1*x1[n] + a2*x2[n]

    obtenemos
        y[n] = T{x[n]} = T{a1*x1[n] + a2*x2[n]} = a1*T{x1[n]} + a2*T{x2[n]}
```

2. El sistema es invariante en el tiempo

Eso significa que si conocemos la respuesta del sistema a una señal de entrada, la respuesta del sistema a la misma señal de entrada desplazada es la respuesta original desplazada la misma cantidad. Es decir,

```
    si tenemos
        y[n] = T{x[n]}

    también se cumple, para todo valor i,
        T{x[n-i]} = y[n-i]
```

3. La señal de entrada es una combinación lineal de impulsos

```
    x[n] = 1*delta[n] + 2*delta[n-1] + 3*delta[n-2] + 4*delta[n-3] +
            4*delta[n-4] + 3*delta[n-5]
```

Juntando estas tres propiedades, tratamos de calcular la respuesta del sistema a la señal x[n]:

```
    y[n] = T{x[n]}

        = T{1*delta[n] + 2*delta[n-1] + 3*delta[n-2] + 4*delta[n-3] +
            4*delta[n-4] + 3*delta[n-5]}

        = 1*T{delta[n]} + 2*T{delta[n-1]} + 3*T{delta[n-2]} + 4*T{delta[n-3]} +
            4*T{delta[n-4]} + 3*T{delta[n-5]}
```

Conocemos el valor de T{delta[n]}, que es h[n] por definición, y por tanto también el valor de T{delta[n-i]}, que es h[n-i]. De esta manera la salida se convierte en una suma de funciones h[n] desplazadas y reescaladas:

```
    y[n] = 1*h[n] + 2*h[n-1] + 3*h[n-2] + 4*h[n-3] + 4*h[n-4] + 3*h[n-5]
```

Gráficamente,

```
        |
        |   *
      * | * |
     -+-+-+-+----------- h[n]
        *

        |     *
        |     |
        *   * |
        |   | |
     ---+-+-+-+--------- 2*h[n-1]
        | |
        | *
```

```
        |       *
        |       |
        |       |
        | *   * |
        | |   | |
        | |   | |
     -----+-+-+-+------- 3*h[n-2]
        |   |
        |   |
        |   *

        |         *
        |         |
        |         |
        |         |
        |   *   * |
        |   |   | |
        |   |   | |
        |   |   | |
     -------+-+-+-+----- 4*h[n-3]
        |     |
        |     |
        |     |
        |     *

        |           *
        |           |
        |           |
        |           |
        |     *   * |
        |     |   | |
        |     |   | |
        |     |   | |
     ---------+-+-+-+--- 4*h[n-4]
        |       |
        |       |
        |       |
        |       *

        |             *
        |             |
        |             |
        |       *   * |
        |       |   | |
        |       |   | |
     -----------+-+-+-+- 3*h[n-5]
        |         |
        |         |
        |         *
```

Suma total (y, por tanto, el resultado de la convolución):

```
        |           *
        |           |
        |       * * |
        |       | | |
        |     * | | |
        |     | | | | *
        |   * | | | | |
        |   | | | | | |
        |   | | | | | |
        | * | | | | | |
      * * | | | | | | |
     -+-+-+-+-+-+-+-+-+- y[n]

    y[n] = [1, _1_, 2, 5, 7, 9, 9, 11, 6]
    donde
        -1 <= n <= 7
```

Los límites de la convolución van desde n=-1 hasta n=7.

Se podían haber calculado previamente, porque los límites van desde (límite inferior de x[n] + límite inferior de h[n]) hasta (límite superior de x[n] + límite superior de h[n]), es decir, desde (0 + -1 = -1) hasta (5 + 2 = 7).

Método 2 ("método analítico")

En este método para calcular la convolución x[n]*h[n] vamos utilizar directamente la definición matemática de la operación:

```
    y[n] = x[n]*h[n] = sumatorio(desde k=-infinito; hasta infinito; x[k]h[n-k])
```

Como sabemos que x[n] está definido entre n=0 y n=5, el sumatorio queda reducido a los valores 0 <= k <= 5.

Como sabemos los límites de la convolución (desde límite_inferior(x)+límite_inferior(h) hasta límite_superior(x)+límite_superior(h)), podemos limitar también los valores de n: -1 <= n <= 7

Expresamos las ecuaciones:

```
    y[-1] = x[0]h[-1-0] +   x[1]h[-1-1] +   x[2]h[-1-2] +   x[3]h[-1-3] +
            x[4]h[-1-4] +   x[5]h[-1-5]
    y[0]  = x[0]h[0-0] +    x[1]h[0-1] +    x[2]h[0-2] +    x[3]h[0-3] +
            x[4]h[0-4] +    x[5]h[0-5]
    y[1]  = x[0]h[1-0] +    x[1]h[1-1] +    x[2]h[1-2] +    x[3]h[1-3] +
            x[4]h[1-4] +    x[5]h[1-5]
    y[2]  = x[0]h[2-0] +    x[1]h[2-1] +    x[2]h[2-2] +    x[3]h[2-3] +
            x[4]h[2-4] +    x[5]h[2-5]
    y[3]  = x[0]h[3-0] +    x[1]h[3-1] +    x[2]h[3-2] +    x[3]h[3-3] +
            x[4]h[3-4] +    x[5]h[3-5]
    y[4]  = x[0]h[4-0] +    x[1]h[4-1] +    x[2]h[4-2] +    x[3]h[4-3] +
            x[4]h[4-4] + x[5]h[4-5]
    y[5]  = x[0]h[5-0] +    x[1]h[5-1] +    x[2]h[5-2] +    x[3]h[5-3] +
            x[4]h[5-4] +    x[5]h[5-5]
```

Como sabemos que h[n] sólo está definido en -1 <= n <= 2 podemos eliminar un montón de términos de las ecuaciones:

```
    y[-1] = x[0]h[-1-0]
    y[0]  = x[0]h[0-0] + x[1]h[0-1]
    y[1]  = x[0]h[1-0] + x[1]h[1-1] + x[2]h[1-2]
    y[2]  = x[0]h[2-0] + x[1]h[2-1] + x[2]h[2-2] + x[3]h[2-3]
    y[3]  = x[1]h[3-1] + x[2]h[3-2] + x[3]h[3-3] + x[4]h[3-4]
    y[4]  = x[2]h[4-2] + x[3]h[4-3] + x[4]h[4-4] + x[5]h[4-5]
    y[5]  = x[3]h[5-3] + x[4]h[5-4] + x[5]h[5-5]
```

Por tanto el resultado de la convolución es,

```
    y[n] = [1*1, (1*-1 + 2*1), (1*1 + 2*-1 + 3*1), ... ]
        = [1, _1_, 2, 5, 7, 9, 9, 11, 6]
    donde
        -1 <= n <= 7
```

Método 3 ("método h reflejada")
--
En este método para calcular la convolución x[n]*h[n] vamos a cambiar ligeramente la definición matemática de la operación:

```
    y[n] = x[n]*h[n] = sumatorio(desde k=-infinito; hasta infinito; x[k]h[n-k])
```

Definimos hr[n] como la reflexión de h[n].

```
    h = [1, _-1_, 1, 2]
    hr = [2, 1, _-1_, 1]
```

Por definición de lo que es la reflexión, se cumple que h[n] = hr[-n].

Reescribimos la ecuación:

```
    y[n] = x[n]*h[n] = x[n]*hr[-n]
        = sumatorio(desde k=-infinito; hasta infinito; x[k]hr[-(n-k)])
        = sumatorio(desde k=-infinito; hasta infinito; x[k]hr[k-n])
```

Tal como está expresado el sumatorio, ahora se puede entender como una multiplicación escalar de los vectores: x[k] y hr[k-n]. La representación gráfica de esta operación es simplemente una señal móvil que se va desplazando por encima de la otra señal y en cada punto multiplicando elemento a elemento y sumando resultados parciales.

Gráficamente:

Para calcular el primer término de y[n] se posiciona la señal hr[n] en el punto más a la izquierda que pueda generar una multiplicación no nula. Eso supone que el último punto a la derecha de hr[n] se posiciona sobre el primero punto de x[n] a la izquierda; en esta posición el borde derecho de la señal hr[] acaba de encontrar el borde izquierda de la señal x[n]:

```
        |
        |
        |     * *
        |   * | | *
        | * | | | |
        * | | | | |
 -------+-+-+-+-+-+-------- x[n]
        |


        |
  *     |
  | *   *
 -+-+-+-+------------------ hr[n-(-1)]
      * |


        |
        |
        *                   resultado
 -------+------------------ y[-1] = 1*1 = 1
        |
```

Para calcular el término y[0]:

```
        |
        |
        |     * *
        |   * | | *
        | * | | | |
        * | | | | |
 -------+-+-+-+-+-+-------- x[n]
        |

        |
    *   |
    | * | *
 ---+-+-+-+---------------- hr[n]
        *

        |
        | *
        | |                 resultado
 -------+-+---------------- y[0] = 1*-1 + 2*1 = 1
        *
```

Para calcular el término y[1]:

```
        |
        |
        |     * *
        |   * | | *
        | * | | | |
        * | | | | |
 -------+-+-+-+-+-+-------- x[n]
        |

        |
      * |
      | *   *
 -----+-+-+-+-------------- hr[n-1]
        | *

        |   *
        |   |
        *   |               resultado
 -------+-+-+-------------- y[1] = 1*1 + 2*-1 + 3*1 = 3
        | |
        | *
```

Y así sucesivamente... Para y[2] multiplicamos:

```
        |
        |
        |     * *
        |   * | | *
        | * | | | |
        * | | | | |
 -------+-+-+-+-+-+-------- x[n]
        |

        |
        *
        | *   *
 -------+-+-+-+------------ hr[n-2]
        |   *
```

Para el término y[3]:

```
        |
        |
        |     * *
        |   * | | *
        | * | | | |
        * | | | | |
 -------+-+-+-+-+-+-------- x[n]
        |

        |
        | *
        | | *   *
 ---------+-+-+-+---------- hr[n-3]
        |     *
```

Para el término y[4]:

```
        |
        |
        |     * *
        |   * | | *
        | * | | | |
        * | | | | |
 -------+-+-+-+-+-+-------- x[n]
        |

        |
        |   *
        |   | *   *
 -----------+-+-+-+-------- hr[n-4]
        |       *
```

Para el término y[5]:

```
        |
        |
        |     * *
        |   * | | *
        | * | | | |
        * | | | | |
 -------+-+-+-+-+-+-------- x[n]
        |

        |
        |     *
        |     | *   *
 -------------+-+-+-+------ hr[n-5]
        |         *
```

Para el término y[6]:

```
        |
        |
        |     * *
        |   * | | *
        | * | | | |
        * | | | | |
 -------+-+-+-+-+-+-------- x[n]
        |

        |
        |       *
        |       | *   *
 ---------------+-+-+-+---- hr[n-6]
        |           *
```

Para el término y[7]:

```
        |
        |
        |     * *
        |   * | | *
        | * | | | |
        * | | | | |
 -------+-+-+-+-+-+-------- x[n]
        |

        |
        |         *
        |         | *   *
 -----------------+-+-+-+-- hr[n-7]
        |           *
```

Al terminar,

```
    y[n] = [1, _1_, 2, 5, 7, 9, 9, 11, 6]
    donde
        -1 <= n <= 7
```

Se puede observar que la señal hr[n] se va deslizando sobre la señal x[n] desde la izquierda de x[n] hasta que desaparece por la derecha.

Se puede observar también que el valor que se está calculando en cada paso, y[i] corresponde al punto donde la señal x[n] está fija y hr[n] está desplazado i posiciones. Es decir, que el origen de coordenadas de hr[n] está desplazado a la posición i.

Comentarios

Aunque los métodos 1 y 3 parecen bastante similares (se pueden ver gráficamente), puedes comprobar que para calcular el término i de la salida, y[i], en el primer método hay que sumar los valores de múltiples gráficas; sin embargo, en el tercer método simplemente hay que desplazar hr[n] a la posición i y hacer una multiplicación entre ambas señales. Decimos que en el primer caso hay una suma en "vertical" mientras que en el segundo caso hay una suma en "horizontal" (simples nombres).