Sistemas y Señales I

Algoritmo de cómputo de la convolución en TD

Temario: Cap. 2: Items 2.2 y 2.4

La respuesta de un Sistema Lineal Estacionario (SLE) caracterizado por su respuesta al impulso $h(n)$ a una entrada arbitraria $u(n)$ viene dada por la **suma de convolución**

$$\boxed{y(n)=\sum_{k=-\infty}^{\infty}u(k)\cdot h(n-k)} \tag{1}$$

Es claro que para implementar un algoritmo que compute esta convolución tanto la respuesta al impulso, como la entrada deben ser de longitud finita. Supongamos que el sistema es causal, es decir $h(n)=0$ para todo $n<0$, y que $h(n)$ es de longitud finita N. Supongamos además que la señal de entrada es causal, es decir $u(n)=0$ para todo $n<0$, y de longitud finita M. La ecuación (1) resulta entonces

$$\boxed{y(n)=\sum_{k=0}^{M-1}u(k)\cdot h(n-k)} \tag{2}$$

Escribamos (2) explícitamente para $n = 0, 1, 2, \ldots$, y $M=N=4$

$$y(0) = h(0)u(0)$$
$$y(1) = h(1)u(0) + h(0)u(1)$$
$$y(2) = h(2)u(0) + h(1)u(1) + h(0)u(2)$$
$$y(3) = h(3)u(0) + h(2)u(1) + h(1)u(2) + h(0)u(3)$$
$$y(4) = 0 \times u(0) + h(3)u(1) + h(2)u(2) + h(1)u(3) + h(0) \times 0$$
$$y(5) = 0 \times u(0) + 0 \times u(1) + h(3)u(2) + h(2)u(3) + h(1) \times 0$$
$$y(6) = 0 \times u(0) + 0 \times u(1) + 0 \times u(2) + h(3)u(3) + h(2) \times 0$$

El sistema de ecuaciones puede escribirse en la forma matricial siguiente

$$\underbrace{\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \\ y(6) \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} h(0) & 0 & 0 & 0 \\ h(1) & h(0) & 0 & 0 \\ h(2) & h(1) & h(0) & 0 \\ h(3) & h(2) & h(1) & h(0) \\ 0 & h(3) & h(2) & h(1) \\ 0 & 0 & h(3) & h(2) \\ 0 & 0 & 0 & h(3) \end{bmatrix}}_{H} \times \underbrace{\begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ u(3) \end{bmatrix}}_{u}$$

Vemos que la matriz H es constante a lo largo de las diagonales paralelas a la diagonal principal. Una matriz con esa estructura se denomina matriz **Toeplitz**.

En Matlab puede construirse fácilmente esa matriz con el comando `toeplitz,` que debe invocarse de la forma

```
>> H = toeplitz(c,r);
```

Donde `c` es la primera columna de la matriz y `r` es la primera fila. Notar que la primera columna es el vector h de respuestas al impulso seguido de un vector de ceros de longitud `length(u)-1`, en tanto que la primera fila es un vector cuyo primer componente es la primera componente del vector h seguido de un vector fila de ceros de longitud `length(u)-1`.

Notar que la longitud del vector y resulta igual a:

$$M+N\text{-}1$$

La implementación del algoritmo en un **`function file`** de Matlab resulta

```
function y = convolucion(u,h)
%=================================
% “u” y “h” deben ser vectores columna.
% La función devuelve en “y” la convolución
% entre “u” y “h”
%==================================
c=[h;zeros(length(u)-1,1)];
r=[h(1) zeros(1,length(u)-1)];
H=toeplitz(c,r);
y=H*u;
```

Algoritmo de cómputo de la correlación en TD

Considerando que la correlación de dos señales puede escribirse en función de la convolución, de la forma:

$$r_{xy}(\ell) = x(\ell) * y(-\ell)$$

puede modificarse fácilmente la función `convolucion` que desarrollamos, para calcular la correlación de dos señales. Debemos reflejar a la señal y respecto al eje de ordenadas. Como la señal $y(\ell)$ la representamos con un vector columna, podemos usar la función de Matlab `flipud` para obtener $y(-\ell)$

```
function rxy = correlacion(x,y)
%=====================================
% "x" e "y" deben ser vectores columna.
% La función devuelve en "rxy" la correlación
% entre "x" e "y"
%======================================
yy=flipud(y);
c=[yy;zeros(length(x)-1,1)];
r=[yy(1) zeros(1,length(x)-1)];
H=toeplitz(c,r);
rxy=H*x;
```