Apunte teórico

Obtención del h mediante `filter` para sistemas en diferencias con *condiciones finales de reposo*

1. Una propiedad útil de la convolución

1.1. Enunciado

Convolucionar dos señales reflejadas equivale a reflejar su convolución

$$x_R(n) \stackrel{\text{def}}{=} x(-n)$$
$$y_R(n) \stackrel{\text{def}}{=} y(-n)$$

$$(x * y)_{(-n)} = (x_R * y_R)_{(n)}$$

O, siendo más laxos con la notación

$$(x * y)_{(-n)} = x_{(-n)} * y_{(-n)}$$

- ¿Qué implica esto para los sistemas LTI?

Si una de las señales que se convolucionan es la respuesta al impulso h de un sistema LTI y la otra la entrada, la propiedad nos dice

$$y = x * h$$
$$y(-n) = (x * h)_{(-n)} = x_{(-n)} * h_{(-n)}$$

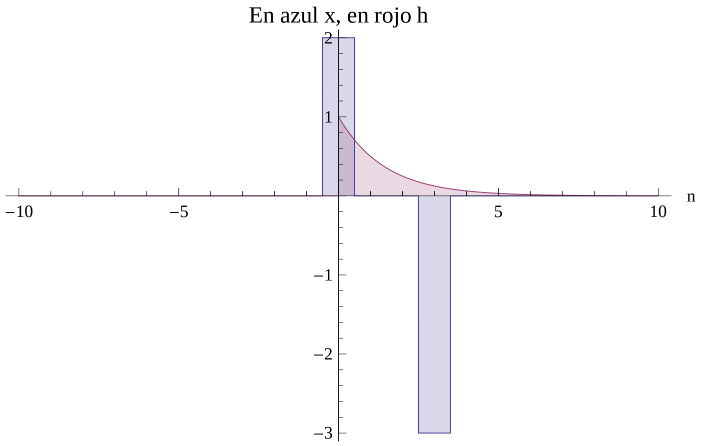
Es decir, reflejar la salida equivale a pasar la entrada reflejada por el sistema LTI reflejado.

1.2. Ejemplo

Convolucionemos dos señales cualesquiera y veamos si se cumple

- *Defino x y h*

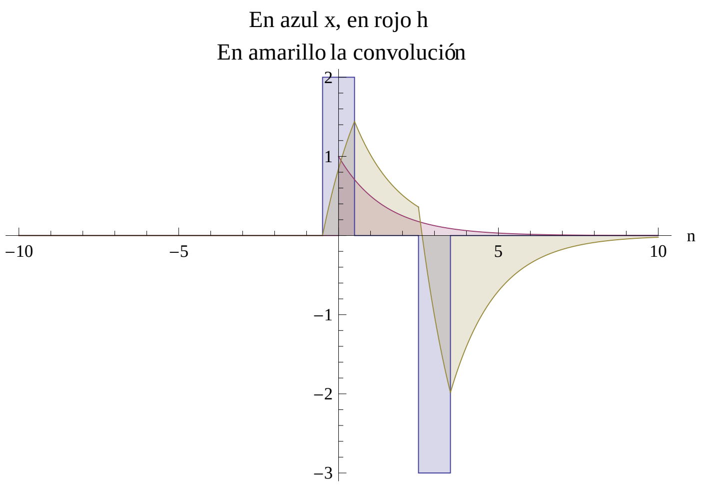
```
ωr = UnitBox; u = UnitStep;
pOptions = Sequence[PlotRange → Full,
   PlotStyle → PointSize[Large], AxesLabel → {"n"}, Filling → Axis];

x[t_] := 2 ωr[t] - 3 ωr[t - 3]
h[t_] := u[t] (1/2)^t
Plot[{x[t], h[t]}, {t, -10, 10}, PlotLabel → "En azul x, en rojo h", Evaluate@pOptions]
```



- *Las convoluciono*

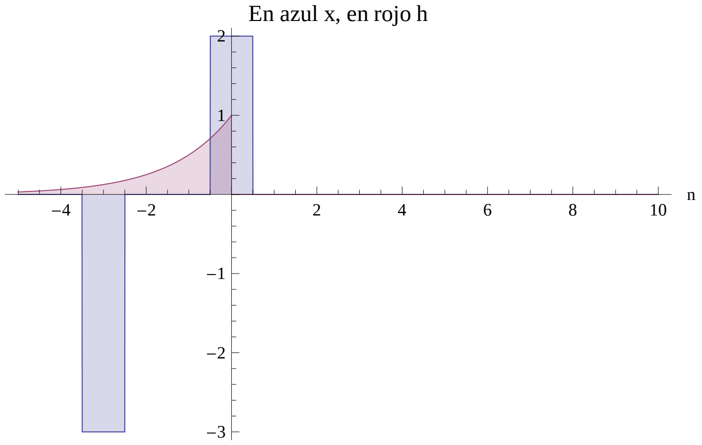
```
y[t_] = Convolve[x[τ], h[τ], τ, t];
```

```
Plot[{x[t], h[t], y[t]}, {t, -10, 10},
 PlotLabel → "En azul x, en rojo h\n En amarillo la convolución",
 Evaluate@pOptions, PlotPoints → 1000]
```



- *Ahora, reflejo la entrada y la respuesta al impulso*

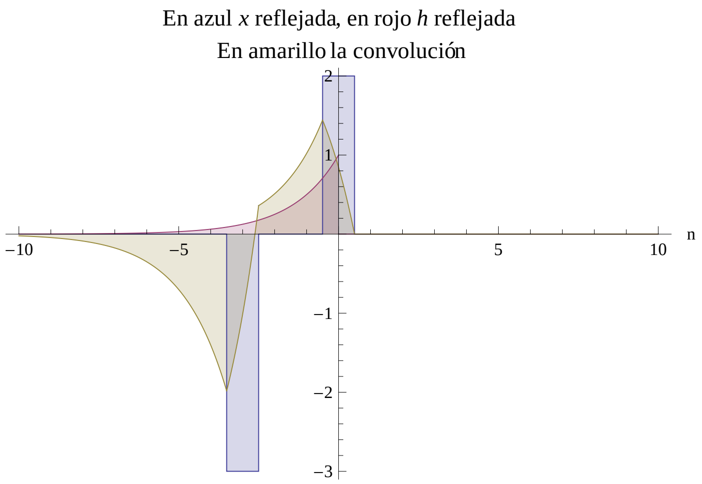
```
Plot[{x[-t], h[-t]}, {t, -5, 10}, PlotLabel → "En azul x, en rojo h", Evaluate@pOptions]
```



- *Las convoluciono*

```
yr[t_] = Convolve[x[-τ], h[-τ], τ, t];
```

```
Plot[{x[-t], h[-t], yr[t]}, {t, -10, 10},
 PlotLabel → "En azul x reflejada, en rojo h reflejada\n En amarillo la convolución",
 Evaluate@pOptions, PlotPoints → 500]
```



Se puede ver que la salida del sistema reflejado $h(-t)$ a la entrada reflejada $x(-t)$ fue la salida del sistema original reflejada $y(-t)$

1.3. Demostración

Recordemos la definición de la convolución

$$(x * y)_{(t)} \overset{\text{def}}{=} \int_{-\infty}^{\infty} x(\tau)\, y(t-\tau)\, d\tau$$

Entonces

$$(x * y)_{(-t)} = \int_{-\infty}^{\infty} x(\tau)\, y((-t)-\tau)\, d\tau$$

Cambio de variable $\tau' = -\tau$

$$= \int_{\infty}^{-\infty} x(-\tau')\, y(-t+\tau')\, d(-\tau')$$

Se cambiaron los límites de integración porque $\tau \longrightarrow \pm\infty$ cuando $\tau' \longrightarrow \mp\infty$

$$= -\int_{\infty}^{-\infty} x(-\tau')\, y(-(t-\tau'))\, d\tau'$$

Puedo invertir los límites de integración multiplicando por -1. Definiendo

$$x_R(t) \overset{\text{def}}{=} x(-t)$$
$$y_R(t) \overset{\text{def}}{=} y(-t)$$

Se tiene

$$= \int_{-\infty}^{\infty} x_R(\tau')\, y_R(t-\tau')\, d\tau' = (x_R * y_R)_{(t)}$$

$$(x * y)_{(-t)} = (x_R * y_R)_{(t)}$$

2. Aplicación a sistemas definidos por ecuaciones en diferencias

Sea un sistema definido por ecuaciones en diferencias de la siguiente forma

$$y(n) + a_1\, y(n-1) + \ldots + a_N\, y(n-N) = b_0\, x(n) + \ldots + b_M\, x(n-M)$$

con *condiciones iniciales de reposo*

- Es un sistema LTI

Es LTI porque las condiciones son iniciales de reposo. Referirse al apunte de ecuaciones en diferencias.

- Es un sistema causal

Escribamos la respuesta al impulso, remplazando $x \to \delta$ y $y \to h$

$$h(n) = b_0\, \delta(n) + \ldots + b_M\, \delta(n-M) - a_1\, h(n-1) - \ldots - a_N\, h(n-N) \quad (1)$$

Se tiene que

$$\forall\, n < 0$$
$$h(n) = 0$$

pues los términos $b_k\, \delta(n-k)$ van a corresponder a la delta evaluada en tiempos negativos, y por condiciones iniciales de reposo los términos dependientes de h van a ser nulos hasta que algún otro término haga que esto cambie.

- Encontremos la ecuación en diferencias del sistema anticausal definido por $h_R(n) \stackrel{\text{def}}{=} h(-n)$

Teníamos la ecuación (1) que definía recursivamente h con condiciones iniciales de reposo. Una ecuación en diferencias que se satisface para $h(-n)$ es entonces

$$h_R(n) = h(-n) = b_0\, \delta(-n) + \ldots + b_M\, \delta(-n-M) - a_1\, h(-n-1) - \ldots - a_N\, h(-n-N)$$
$$= b_0\, \delta(-n) + \ldots + b_M\, \delta(-n-M) - a_1\, h_R(-(-n-1)) - \ldots - a_N\, h_R(-(-n-N))$$

Desarrollando y usando que δ es una señal par

$$h_R(n) = b_0\, \delta(n) + \ldots + b_M\, \delta(n+M) - a_1\, h_R(n+1) - \ldots - a_N\, h_R(n+N)$$

Es decir, la ecuación en diferencias siguiente satisface que ante un impulso puede devolver h_R

$$y(n) = b_0\, x(n) + \ldots + b_M\, x(n+M) - a_1\, y(n+1) - \ldots - a_N\, y(n+N)$$

Ahora bien, h_R es una señal a izquierda, pues h era a derecha. Las condiciones de contorno para obtener h_R como respuesta al impulso ahora se convirtieron entonces en *condiciones finales de* *reposo*

- Resumiendo

Dado el sistema con *condición inicial de reposo*

$$y(n) + \ldots + a_N\, y(n-N) = b_0\, x(n) + \ldots + b_M\, x(n-M)$$

El sistema reflejado se obtuvo “cambiando de signo” los argumentos

$$y(n) + \ldots + a_N\, y(n+N) = b_0\, x(n) + \ldots + b_M\, x(n+M)$$

y cambiando la condición a *condición final de reposo*

Intuitivamente, se cambió el sentido del tiempo:

- De iterar desde tiempos pasados a futuros arrancando de una condición nula se pasó a iterar desde tiempos futuros a pasados comenzando de una condición final nula
- Todos los desplazamientos temporales de la relación de la ecuación en diferencias cambiaron de lado: los desplazamientos a derecha pasaron a ser a izquierda y viceversa.

3. Aplicación: utilizando `filter` para un sistema finales de reposo

Se tiene por ejemplo la siguiente ecuación en diferencias

$$y(n) = \frac{1}{2}\, y(n-1) + 0.2\, x(n)$$

- Respuesta al impulso con condiciones iniciales de reposo

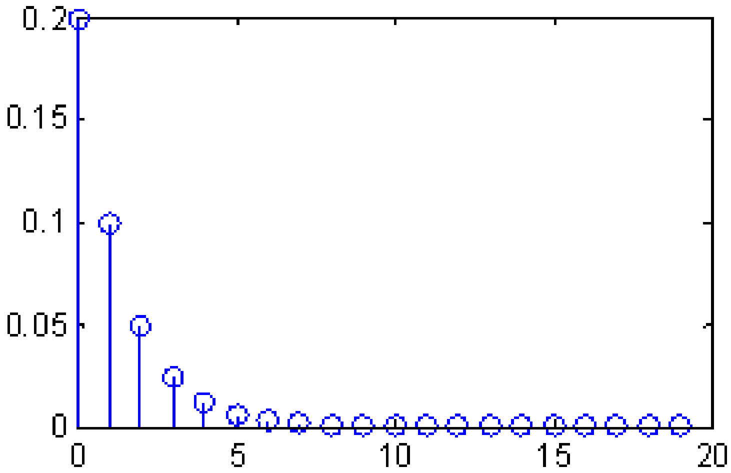
Los primeros 20 puntos de la salida al sistema con *condiciones iniciales de reposo* los podemos obtener de forma directa usando `filter`

Se tiene

$$y(n) - \frac{1}{2}\, y(n-1) = 0.2\, x(n)$$

```
y = filter ([0.2], [1, -0.5], [1, zeros (1, 19)]);
stem (0 : 19, y)
```

MATLAB



- Respuesta al impulso con condiciones finales de reposo

Utilicemos los resultados obtenidos anteriormente. Se tiene

$$y(n) - \frac{1}{2}\, y(n-1) = 0.2\, x(n)$$

“Invirtiendo el tiempo” obtenemos otra ecuación en diferencias

$$y(n) - \frac{1}{2}\, y(n+1) = 0.2\, x(n)$$

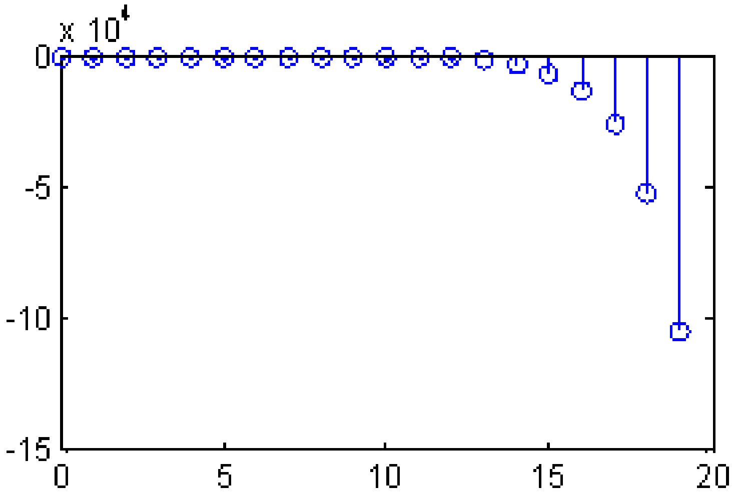
Reescribámosla de una forma que le guste a `filter`

$m = n + 1 \quad \Longrightarrow \quad n = m - 1$

$$-\frac{1}{2}\, y(m) + y(m-1) = 0.2\, x(m-1)$$

Entonces, la salida del sistema definido por la ecuación anterior con condiciones iniciales de reposo, ante la entrada reflejada, es la salida que deseamos, reflejada. El impulso es par, así que es invariante ante la reflexión.

```
yr = filter ([0, 0.2], [-0.5, 1], [1, zeros (1, 19)]);
stem (0 : 19, yr)
```
MATLAB



Ahora bien, esta es la salida al sistema reflejado frente a la entrada reflejada. Lo que buscamos, dada la propiedad anterior, es esta señal reflejada.

Es sólo cuestión de interpretar el vector devuelto por el *MATLAB* de la forma apropiada

```
stem (-19 : 0, yr (20 : -1 : 1))
```
MATLAB