Filtros Digitales 1

Juan-Pablo Cáceres
CCRMA
Stanford University

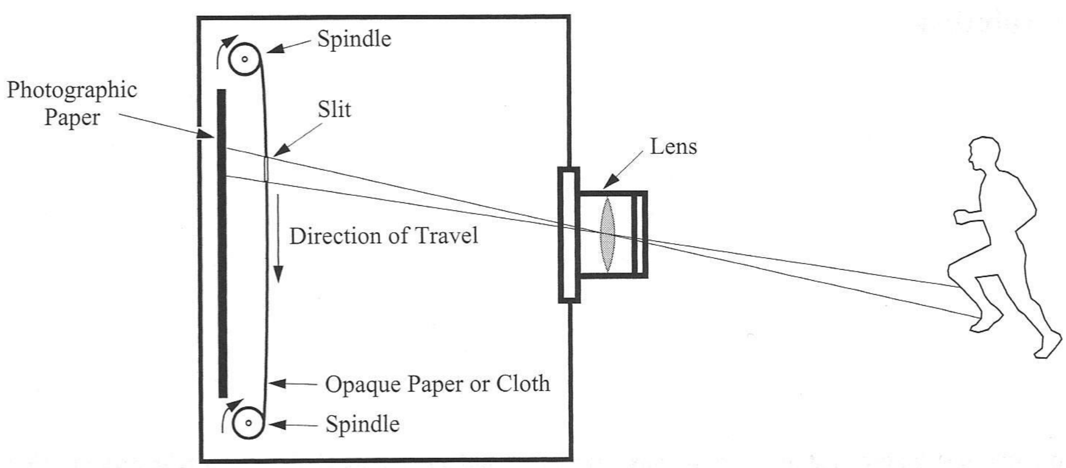
Agosto, 2007

Contents

Convolution

Jacques Henri Lartigue, 1913

Imagen Convolution



Essentials of **Convolution**:
Time Shifting and **Combining** signals.

Why is Convolution important?

Any **Linear Time Invariant System** can be represented by a convolution.

$$x[n] \longrightarrow \boxed{\;h\;} \longrightarrow y[n]$$

$$y = h * x$$

h is a **Filter**, that is, something that somehow **changes** x and **transforms** it into y

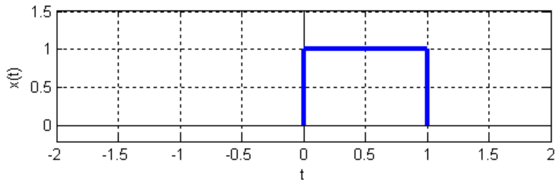
Convolution in Detail

$$\boxed{y[n] = \sum_{k=0}^{n} h[k]x[n-k]} \quad \textbf{Convolution}$$

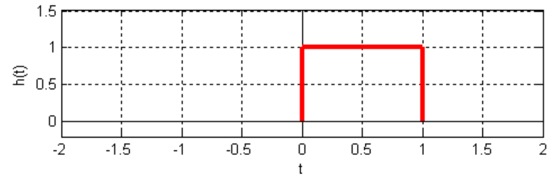
$$
\begin{aligned}
n = 0 &\qquad y[0] = h[0]x[0] \\
n = 1 &\qquad y[1] = h[0]x[1] + h[1]x[0] \\
n = 1 &\qquad y[2] = h[0]x[2] + h[1]x[1] + h[2]x[0] \\
n = 1 &\qquad y[3] = h[0]x[3] + h[1]x[2] + h[2]x[1] + h[3]x[0] \\
&\qquad \ldots
\end{aligned}
$$

An example of Convolution

We take two signals $x(t)$ and $h(t)$ and calculate $y(t) = h(t) * x(t)$,

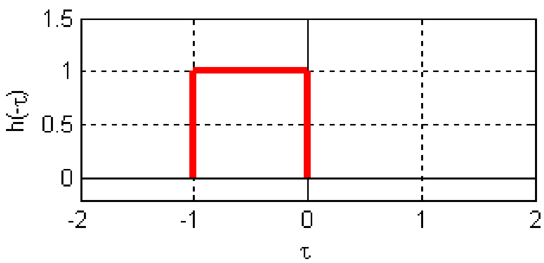


$x(t)$

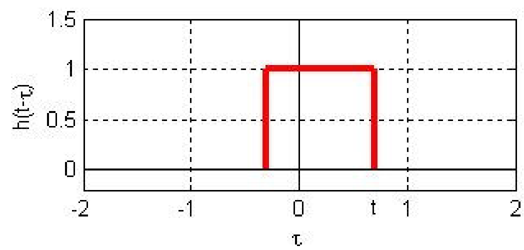


$h(t)$

First we mirror $h(\tau) \to h(-\tau)$, and we move it $\to h(t-\tau)$,



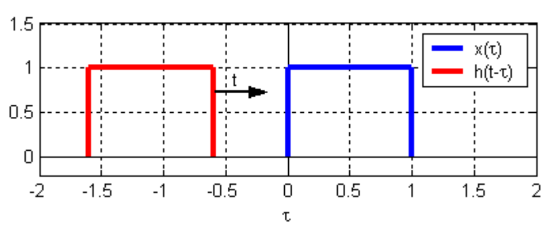
$h(-\tau)$

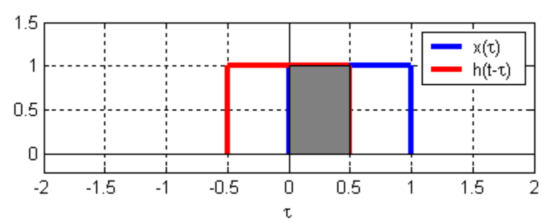


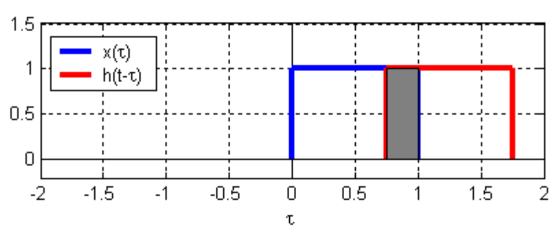
$h(t-\tau)$

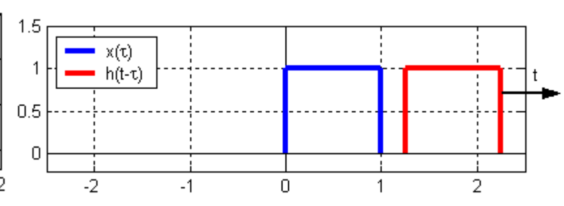
An example of Convolution

After that we “shift” $h(t)$ towards $x(t)$, and we multiply each time...

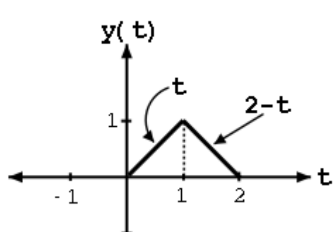






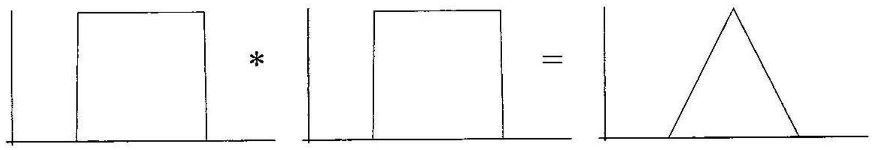


...and we get this as a result:

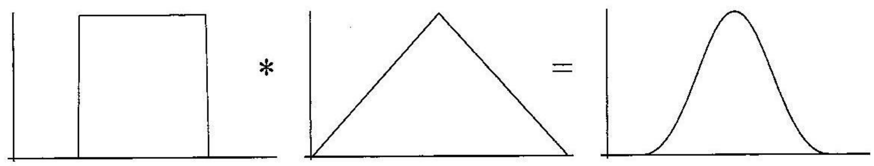


Other Examples

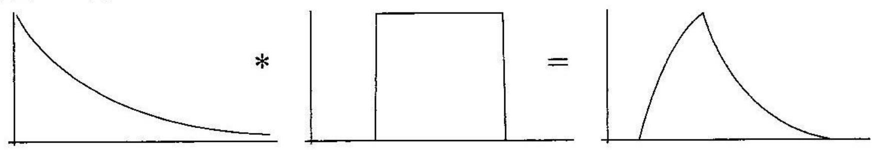
$f(x) * g(x) = h(x)$



$f(x) * g(x) = h(x)$

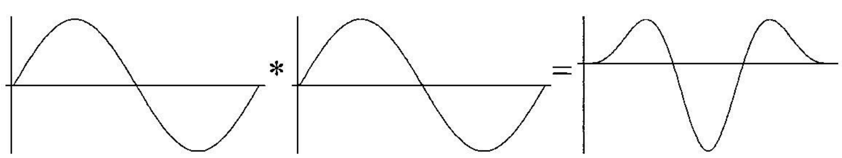


$f(x) * g(x) = h(x)$

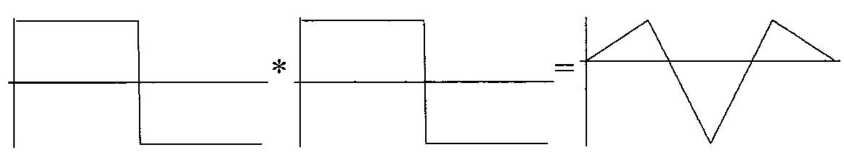


Other Examples

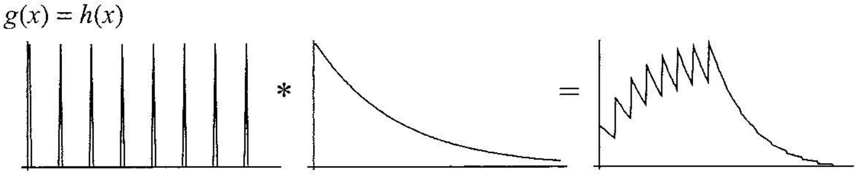
$f(x) * g(x) = h(x)$



$f(x) * g(x) = h(x)$

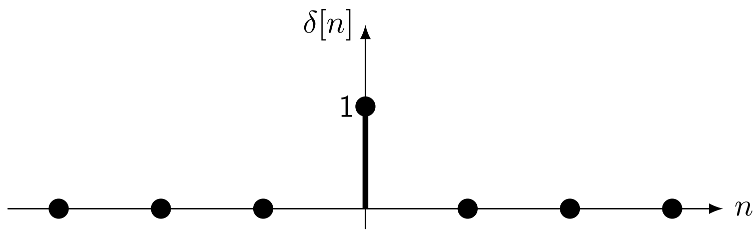


$f(x) * g(x) = h(x)$



Unitary Impulse Function

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad \textbf{Unitary Impulse Function}$$



The $\delta[n]$ function is the identity operator for convolution, i.e.:

$$\delta[n] * x[n] = x[n]$$

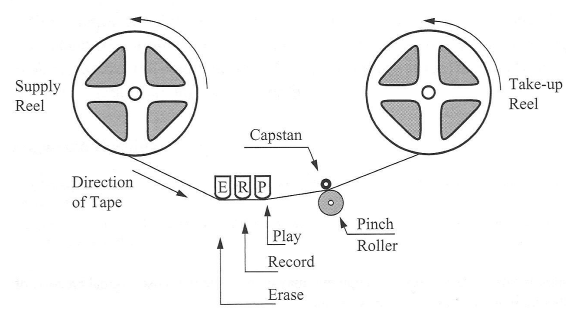
$$\delta[n - n_o] * x[n] = x[n - n_o]$$

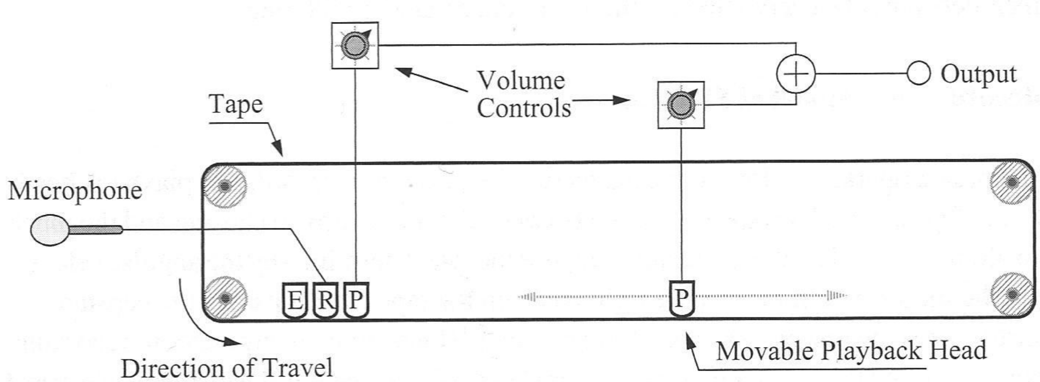
Convolution is a linear operation, which implies that:

$$(A\delta[n - n_o]) * x[n] = Ax[n - n_o]$$

Filters and Delay

Echoplex: echo machine.





An Echoplex as a Filter

If we try with sines of different frequencies:

- Some frequencies sound louder
- Others are nearly gone

Why?

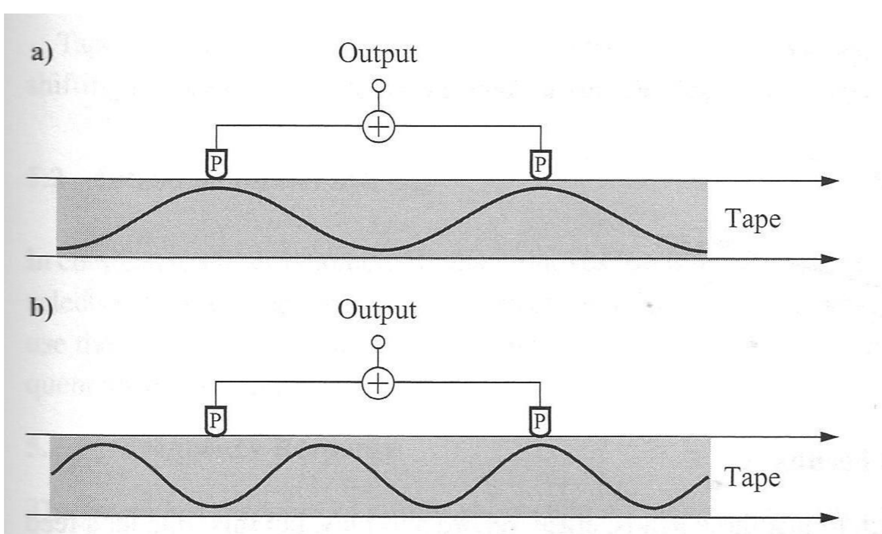
An Echoplex as a Filter

If we try with sines of different frequencies:

- Some frequencies sound louder
- Others are nearly gone

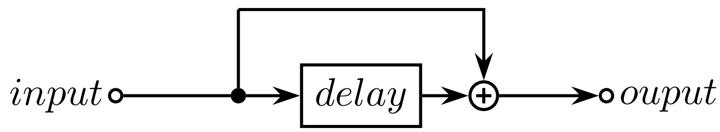
Why?

Some frequencies are **added**, while others are **cancelled** out.



A model of the Echoplex

We can model an Echoplex with 2 echoes with the following block diagram,

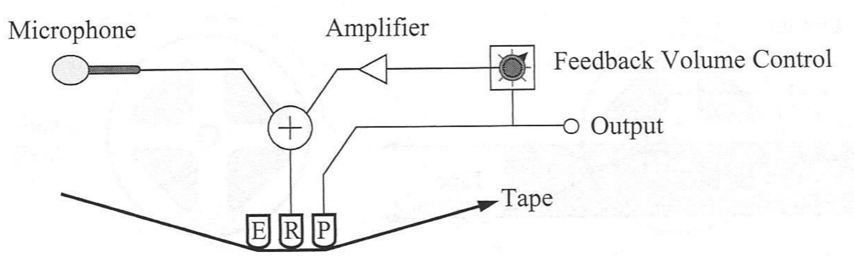


This is a particular configuration of a generic FIR filter (Finite Impulse Response Filter).

This system creates two echoes only, and that's it.

Tape Recorder with Feedback

If we create a model similar to the previous one but we add **feedback**

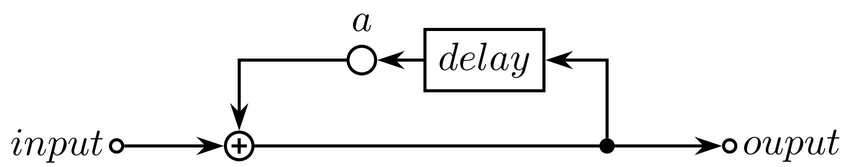


This system has two new characteristics compared with the previous one:

- Its output never stops (in theory or ideal situations)
- Some settings of the *"Feedback Volume Control"* are **unstable**, i.e., the output volume grows without bounds

Model of a Tape Recorder with Feedback

We can model our Echoplex with Feedback withe the following block diagram wich include an element for **feedback**,

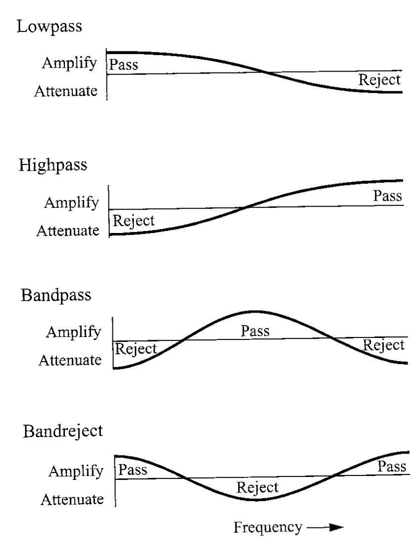


This is a particular configuration of an IIR filter (Infinite Impulse Response Filter).

This system can generate echoes forever.

Basic Filter Types

- *Lowpass*
- *Highpass*
- *Bandpass*
- *Bandreject*
- *Allpass* only changes the **phase** of the signal

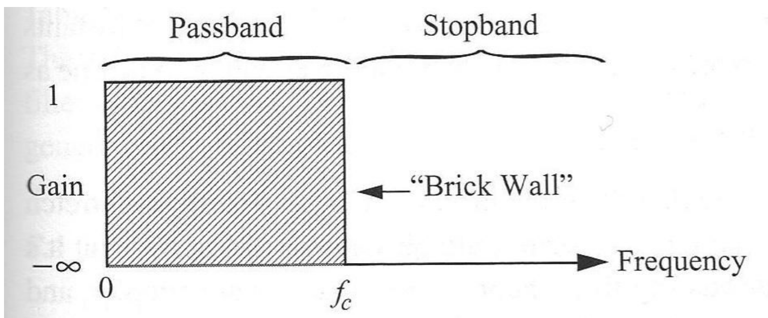


We can use this basic filter types as building blocks for more complex filters.

Frequency Response

$$\boxed{H(k) = \frac{O(k)}{I(k)}} \textit{ Respuesta en Frecuencia}$$

- $O(k)$: Output Frequency Response
- $I(k)$: Input Frequency Response



Ideal Frequency Reponse of a *Lowpass Filter*