Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI

Lecture 10: Linearity and Time Invariance

ECE 401: Signal and Image Analysis

University of Illinois

2/28/2017



▲□▶ ▲圖▶ ▲臣▶ ▲臣▶ = 臣 = のへで

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

1 Convolution Review







4 Convolution Works IFF LTI

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
0	0000	0000	0
Outline			

1 Convolution Review

2 Linearity

3 Time Invariance



★□> <圖> < E> < E> E のQ@

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
•	0000	0000	O
Convolution			

Find y[n] = x[n] * h[n] using graphical convolution, where

$$x[n] = \begin{cases} 1 & -1 \le n \le 1\\ 0 & \text{otherwise} \end{cases}$$
$$h[n] = \delta[n] - \delta[n-1]$$

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
0	0000	0000	0
Outline			









▲□▶ ▲□▶ ▲臣▶ ▲臣▶ 三臣 - のへぐ

Convolution Review Linearity oooo Convolution Works IFF LTI Oooo Convolution Works IFF LTI Oooo

Suppose, when you put $x_k[n]$ into some system, $y_k[n]$ is the signal that comes out, for $1 \le k \le 3$. Then the system is **linear** if and only if

 $x_3[n] = ax_1[n] + bx_2[n] \Leftrightarrow y_3[n] = ay_1[n] + by_2[n]$

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
O	0000	0000	0
Example			

$$y[n] = x^2[n]$$

$$y_3[n] = x_3^2[n] = a^2 x_1^2[n] + 2abx_1[n]x_2[n] + b^2 x_2^2[n]$$

but

$$ay_1[n] + by_2[n] = ax_1^2[n] + bx_2^2[n]$$

◆□ ▶ < 圖 ▶ < 圖 ▶ < 圖 ▶ < 圖 • 의 Q @</p>

These are not equal, so the system is not linear.

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
O	00●0	0000	0
Example			

$$y[n] = nx[n]$$

$$y_3[n] = nx_3[n] = anx_1[n] + bnx_2[n]$$

but

$$ay_1[n] + by_2[n] = anx_1[n] + bnx_2[n]$$

These are equal, so the system is linear.

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
O	000●	0000	O
Example			

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

$$y_3[n] = \sum_{t=-\infty}^n x_3[t] = a \sum_{t=-\infty}^n x_1[t] + b \sum_{t=-\infty}^n x_2[t]$$

but

$$ay_1[n] + by_2[n] = a \sum_{t=-\infty}^n x_1[t] + b \sum_{t=-\infty}^n x_2[t]$$

・ロト ・回 ・ ・ ヨ ・ ・ ヨ ・ うへぐ

These are equal, so the system is linear.

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
O	0000		0
Outline			

1 Convolution Review

2 Linearity

3 Time Invariance

4 Convolution Works IFF LTI

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
0	0000	●000	0
Time Invariance :	= Shifting		

Suppose, when you put $x_k[n]$ into some system, $y_k[n]$ is the signal that comes out, for $1 \le k \le 3$. Then the system is **time-invariant** if and only if

$$x_2[n] = x_1[n-m] \Leftrightarrow y_2[n] = y_1[n-m]$$

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
O	0000	0●00	0
Example			

$$y[n] = x^2[n]$$

$$y_2[n] = x_2^2[n] = x_1^2[n-m]$$

but

$$y_1[n-m] = x_1^2[n-m]$$

◆□ ▶ < 圖 ▶ < 圖 ▶ < 圖 ▶ < 圖 • 의 Q @</p>

These are equal, so the system is time-invariant.

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
O	0000	00●0	0
Example			

$$y[n] = nx[n]$$

$$y_2[n] = nx_2[n] = nx_1[n-m]$$

but

$$y_1[n-m] = (n-m)x_1[n-m]$$

◆□ ▶ < 圖 ▶ < 圖 ▶ < 圖 ▶ < 圖 • 의 Q @</p>

These are not equal, so the system is not time-invariant.

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
0	0000	000●	0
Example			

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

$$y_2[n] = \sum_{t=-\infty}^n x_2[t] = \sum_{t=-\infty}^n x_1[t-m] = \sum_{\tau=-\infty}^{n-m} x_1[\tau]$$

but

$$y_1[n-m] = \sum_{t=-\infty}^{n-m} x_1[t]$$

◆□ ▶ < 圖 ▶ < 圖 ▶ < 圖 ▶ < 圖 • 의 Q @</p>

These are equal, so the system is time-invariant.

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
0	0000	0000	©
Outline			

Convolution Review

2 Linearity

3 Time Invariance



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

Convolution Review	Linearity	Time Invariance	Convolution Works IFF LTI
			•

This system is linear and time invariant:

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

That means we can compute its output, in response to any input, using a convolve function (for example, using np.convolve):

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

To find h[n], we just put $x[n] = \delta[n]$ into the system, and see what comes out. What comes out is

$$h[n] = \begin{cases} 1 & n \ge 0\\ 0 & n < 0 \end{cases}$$

Remember, by the way, that this signal is called a unit step function, and is denoted u[n], thus for this system, h[n] = u[n].

- 日本 - 4 日本 - 4 日本 - 日本