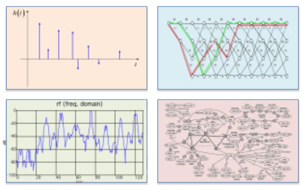


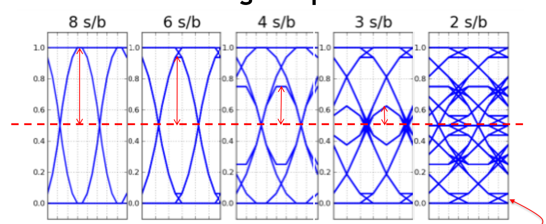
INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

6.02 Spring 2012
Lecture #12

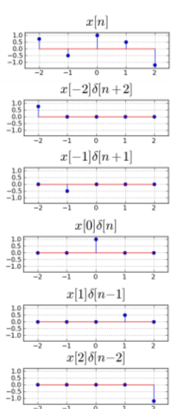
- Convolution
- LTI System Interconnections
- Deconvolution

Choosing Samples/Bit



Given h[n], you can use the eye diagram to pick the number of samples transmitted for each bit (N):

Reduce N until you reach the noise margin you feel is the minimum acceptable value.



Unit Sample Decomposition

A discrete-time signal can be decomposed into a sum of time-shifted, scaled unit sample functions.

Example: in the figure, x[n] is the sum of

x[-2]δ[n+2] + x[-1]δ[n+1] + … + x[2]δ[n-2].

In general:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

For any particular index, only one term of this sum is non-zero

Convolution!

If system S is both linear and time-invariant (LTI), then we can use the **unit sample response h[n]** to predict the response to *any* input waveform x[n]:

Sum of shifted, scaled unit sample functions

Sum of shifted, scaled unit sample *responses*, with the same scale factors

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \longrightarrow \boxed{S \; h[.]} \longrightarrow y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

CONVOLUTION SUM

Indeed, the unit sample response h[n] completely characterizes the LTI system S, so you often see

x[n] → h[.] → y[n]

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

To Convolve (but not to "Convolute"!)

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

A simple graphical implementation:

Plot x[.] and h[.] as a function of the dummy index (k or m above)

Flip (i.e., reverse) one signal in time, **slide** it right by n (slide left if n is –ve), take the **dot.product** with the other.

This yields the value of the convolution at the single time n.

'flip one & slide by n …. dot.product with the other'

Channels as LTI Systems

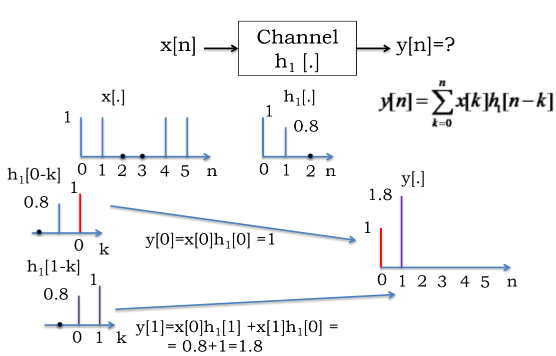
Many transmission channels can be effectively modeled as LTI systems. When modeling transmissions, there are few simplifications we can make:

- We'll call the time transmissions start t=0; the signal before the start is 0. So x[m] = 0 for m < 0.
- Real-word channels are ***causal:*** the output at any time depends on values of the input at only the present and past times. So h[m] = 0 for m < 0.

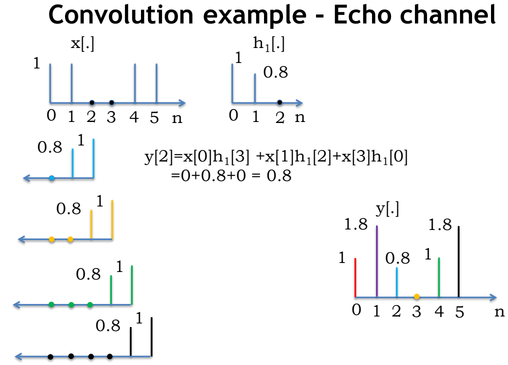
$$y[n]=\sum_{k=-\infty}^{\infty} x[k]h[n-k]=\sum_{k=0}^{\infty} x[k]h[n-k]=\sum_{k=0}^{n} x[k]h[n-k]=\sum_{j=0}^{n} x[n-j]h[j]$$

start at t=0 causal j=n-k

Convolution example - Echo channel



Convolution example - Echo channel



Properties of Convolution

$$(x*h)[n] \equiv \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{m=-\infty}^{\infty} h[m]x[n-m]$$

The second equality above, which follows from the simple change of variables **n-k=m**, establishes that convolution is commutative:

$$x*h = h*x$$

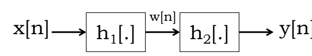
Convolution is associative:

$$h_2*(h_1*x) = (h_2*h_1)*x$$

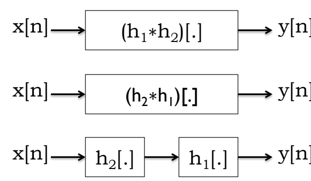
Convolution is distributive:

$$(h_1+h_2)*x = (h_1*x)+(h_2*x)$$

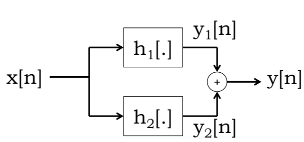
Series Interconnection of LTI Systems



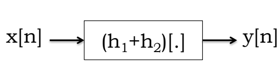
$$y = h_2*w = h_2*(h_1*x) = (h_2*h_1)*x = (h_1*h_2)*x$$



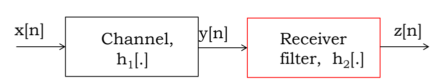
Parallel Interconnection of LTI Systems



$$y = y_1 + y_2 = (h_1*x)+(h_2*x) = (h_1+h_2)*x$$



"Deconvolving" Output of Channel with Echo



Suppose channel is LTI with

$h_1[n]=\delta[n]+0.8\delta[n-1]$

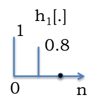
Find $h_2[n]$ such that $z[n]=x[n]$

⇨ $(h_2*h_1)[n]=\delta[n]$

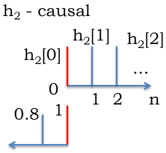
Good exercise in applying Flip/Slide/Dot.Product

Deconvolution example

$h_2[n] = h_2[0]\delta[n] + h_2[1]\delta[n-1] + \cdots + h_2[k]\delta[n-k] + \cdots$



h_2 - causal



Deconvolution goal:
$(h_2 * h_1)[n] = \delta[n]$

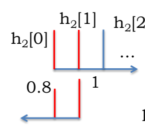
$$1 \cdot h_2[0] = 1$$

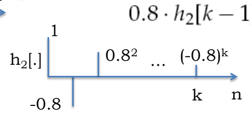
$$0.8 \cdot h_2[0] + 1 \cdot h_2[1] = 0$$

$$0.8 \cdot h_2[1] + 1 \cdot h_2[2] = 0$$

$$\ldots$$

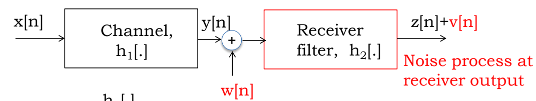
$$0.8 \cdot h_2[k-1] + 1 \cdot h_2[k] = 0$$

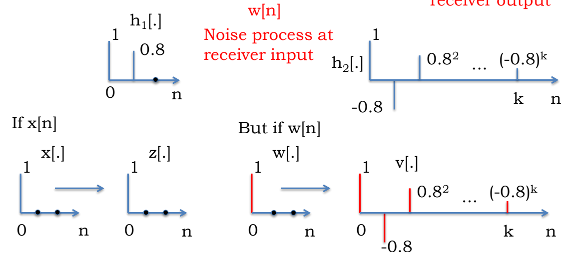




$h_2[k] = (-0.8)^k \delta[k]$

Noise impact on Deconvolver





Noise persists forever!!!

Stability

What ensures that the infinite sum

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

is well-behaved?

One important case: If the unit sample response is *absolutely summable,* i.e.,

$$\sum_{m=-\infty}^{\infty} |h[m]| < \infty$$

and the input is *bounded,* i.e., $|x[k]| \le M < \infty$

Under these conditions, the convolution sum is well-behaved, and the *output* is guaranteed to be ***bounded***.

The absolute summability of h[n] is necessary and sufficient for this bounded-input bounded-output (BIBO) stability.