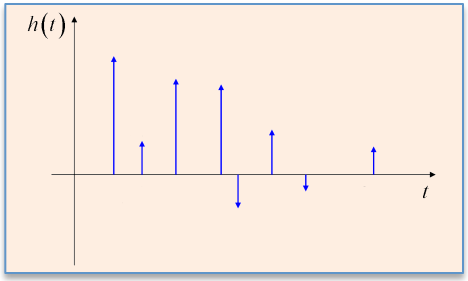
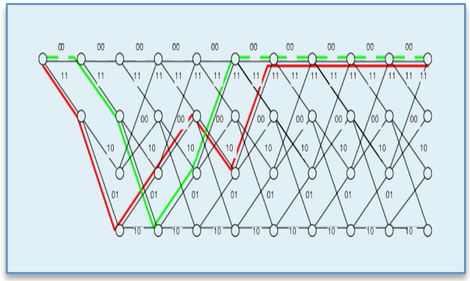
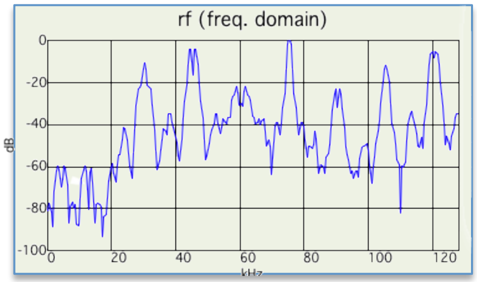
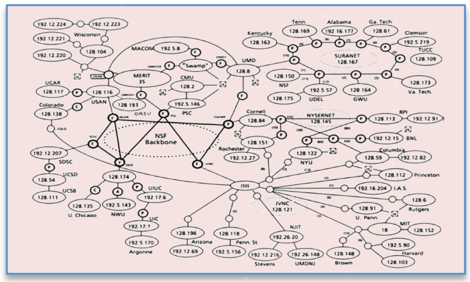








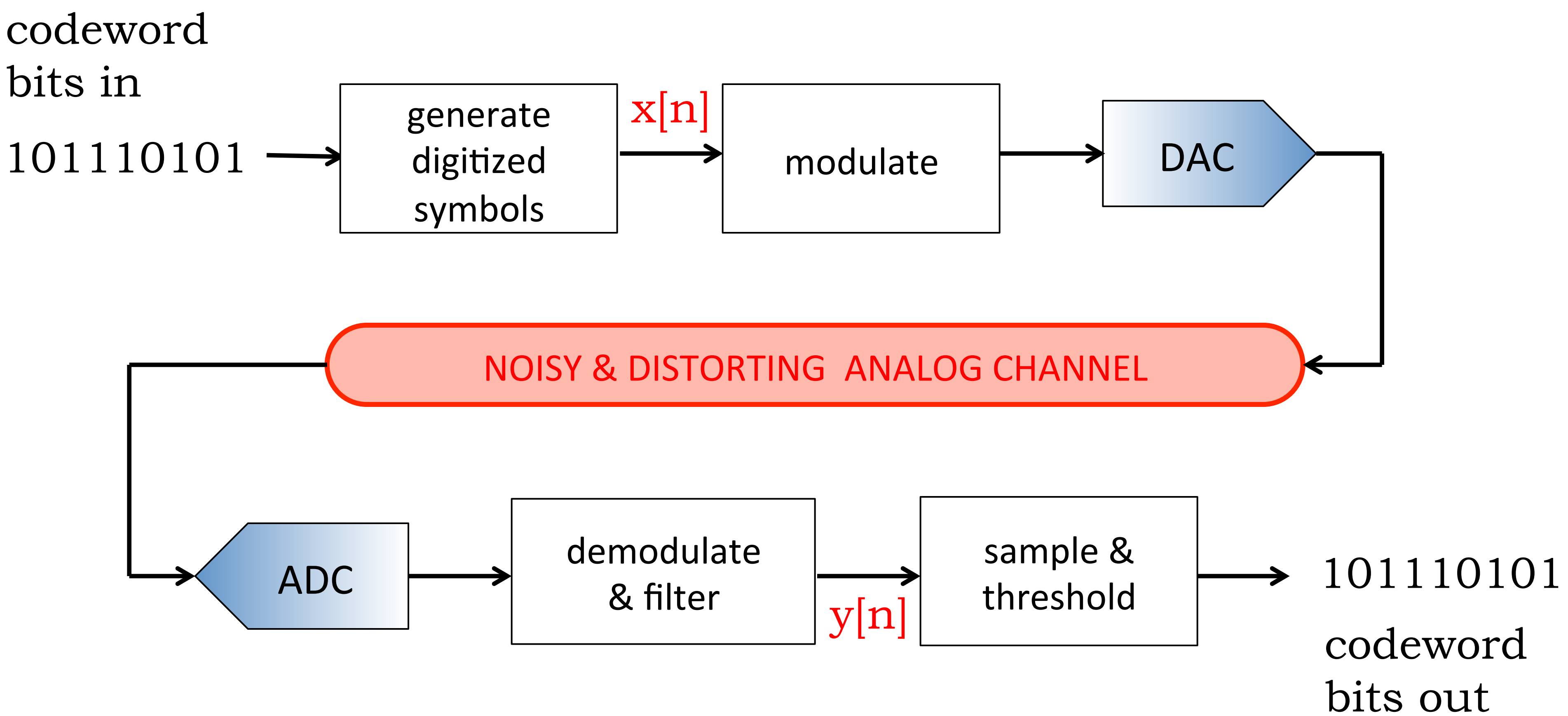
INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

6.02 Fall 2011
Lecture #11

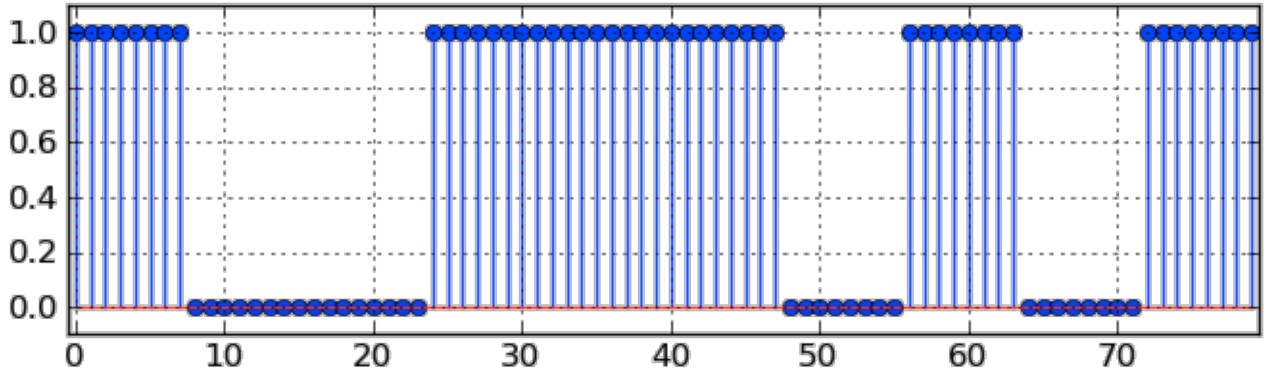
- LTI channel models
- Superposed step responses; eye diagrams
- Convolution
  - definition, properties
  - causality, stability

Modeling Channel Behavior

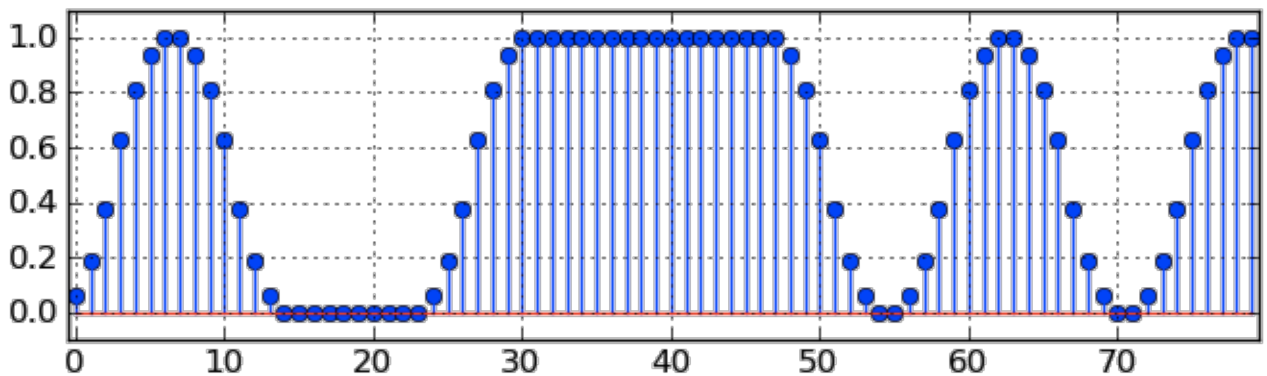


Transmission over a Channel

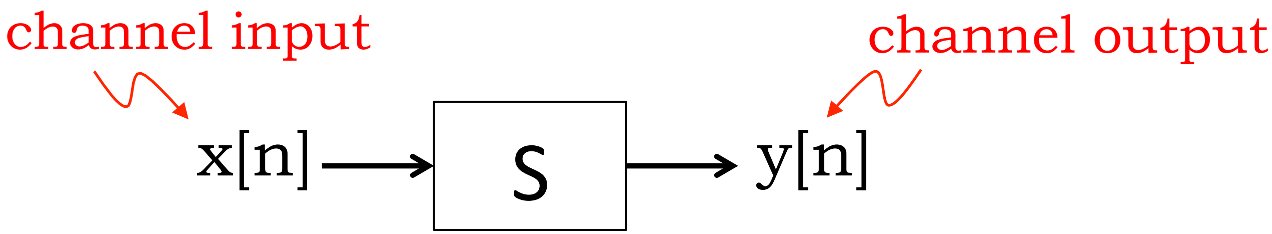
Signal x[n] from digitized symbols at transmitter



Distorted noise-free signal y[n] at receiver



The Baseband* Channel



Starting point:

Try a linear, time-invariant (LTI) model!

Keith's demo from last time suggests this may not be unreasonable for the acoustic channel in this room.

*From before the modulator to after the demodulator, i.e., hiding the modulation/demodulation

Why So Eager for LTI?

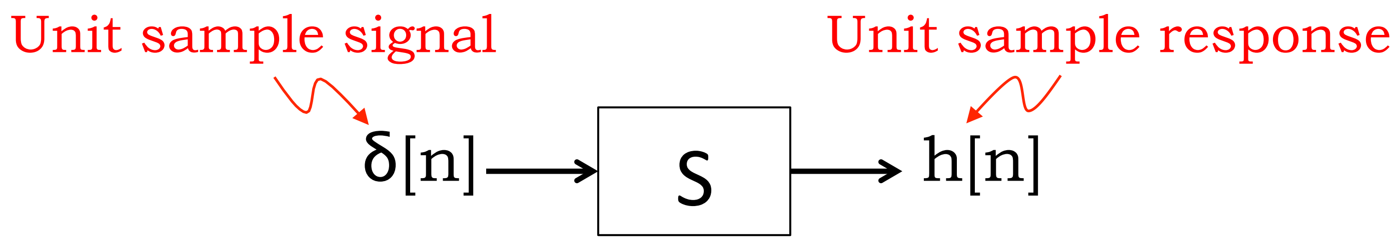
- Lots of structure, mathematically tractable, rich basis for analysis and design

- Good model for small perturbations from a constant equilibrium point (for the same reason that the linear term of a Taylor series is a good local description)

- Even when the overall system may be significantly nonlinear and/or time varying, on short enough time scales the subsystems or modules may often be well approximated as LTI, so LTI design methods form a good starting point

Important to check LTI-based designs against more realistic simulations and analysis before deploying!

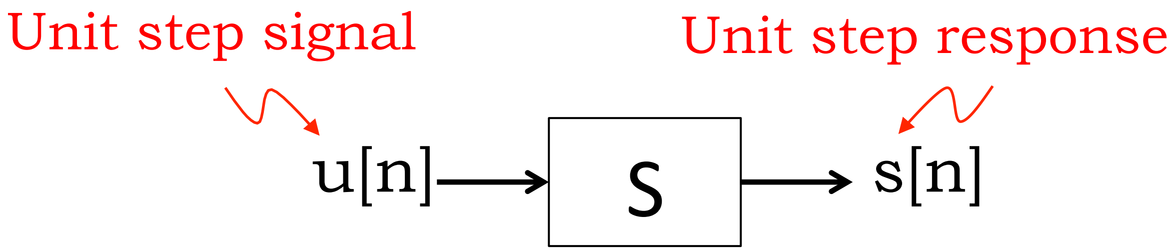
Unit Sample Response



The *unit sample response* of a system S is the response of the system to the unit sample input. We will typically denote the unit sample response as h[n].

Unit Step Response

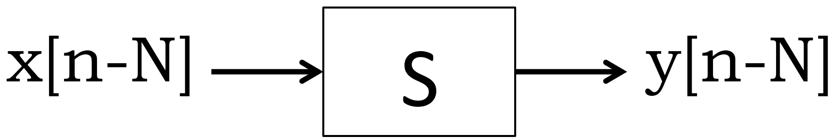
Similarly, the *unit step response* s[n]:



Time Invariant Systems

Let y[.] be the response of S to input x[.]

If for all possible sequences x[n] and integers N



then system S is said to be *time invariant* (TI). A time shift in the input sequence to S results in an identical time shift of the output sequence.

In particular, for a TI system, a shifted unit sample function $\delta[n-N]$ at the input generates an identically shifted unit sample response $h[n-N]$ at the output.

Similarly, $u[n-N]$ generates $s[n-N]$.

Linear Systems

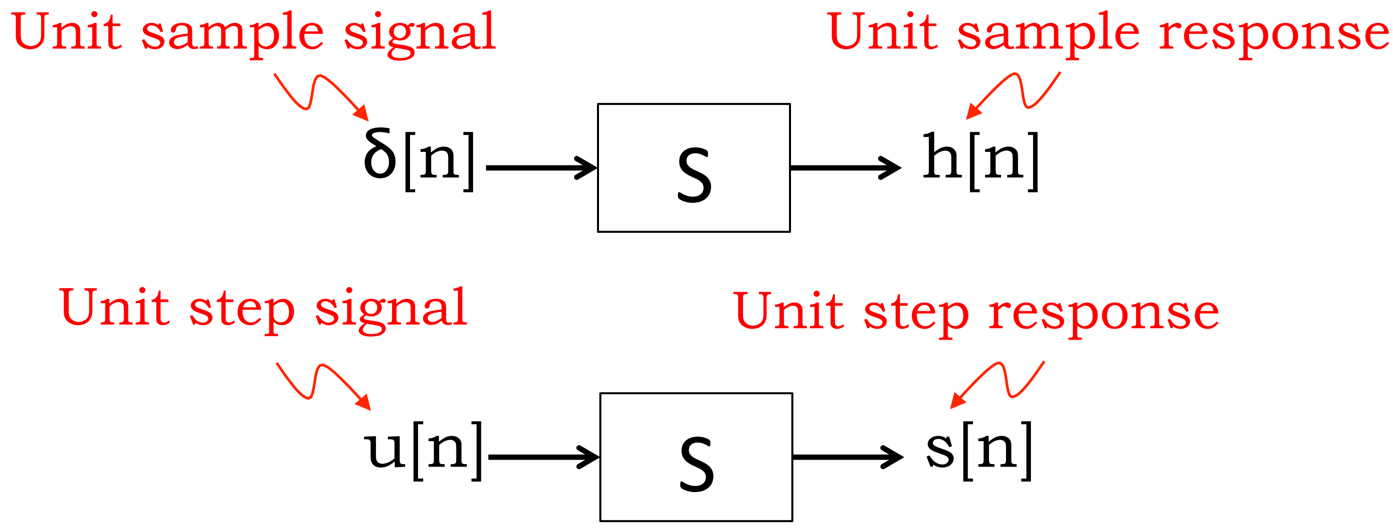
Let $y_1[.]$ be the response of S to an arbitrary input $x_1[.]$, and $y_2[.]$ be the response to an arbitrary input $x_2[.]$

If the response to linear combinations of these two inputs equals the same linear combination of the respective individual responses, then system S is said to be *linear* (L):

$$ax_1[n]+bx_2[n] \longrightarrow \boxed{S} \longrightarrow ay_1[n]+by_2[n]$$

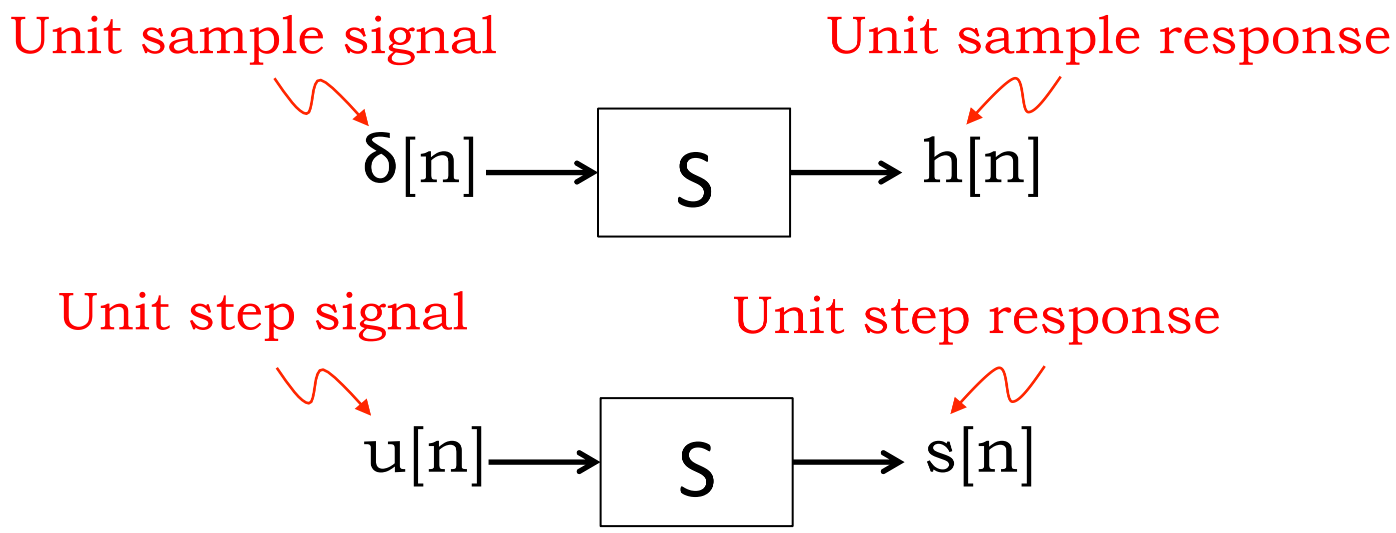
More generally, if the input is the weighted sum of several signals, the response of a linear system is the corresponding *superposition* of the respective responses to those signals (i.e., the weighted sum of these responses, using the *same* weights as in the input).

Relating h[n] and s[n] of an LTI System



$$\delta[n] = u[n] - u[n-1]$$

Relating h[n] and s[n] of an LTI System



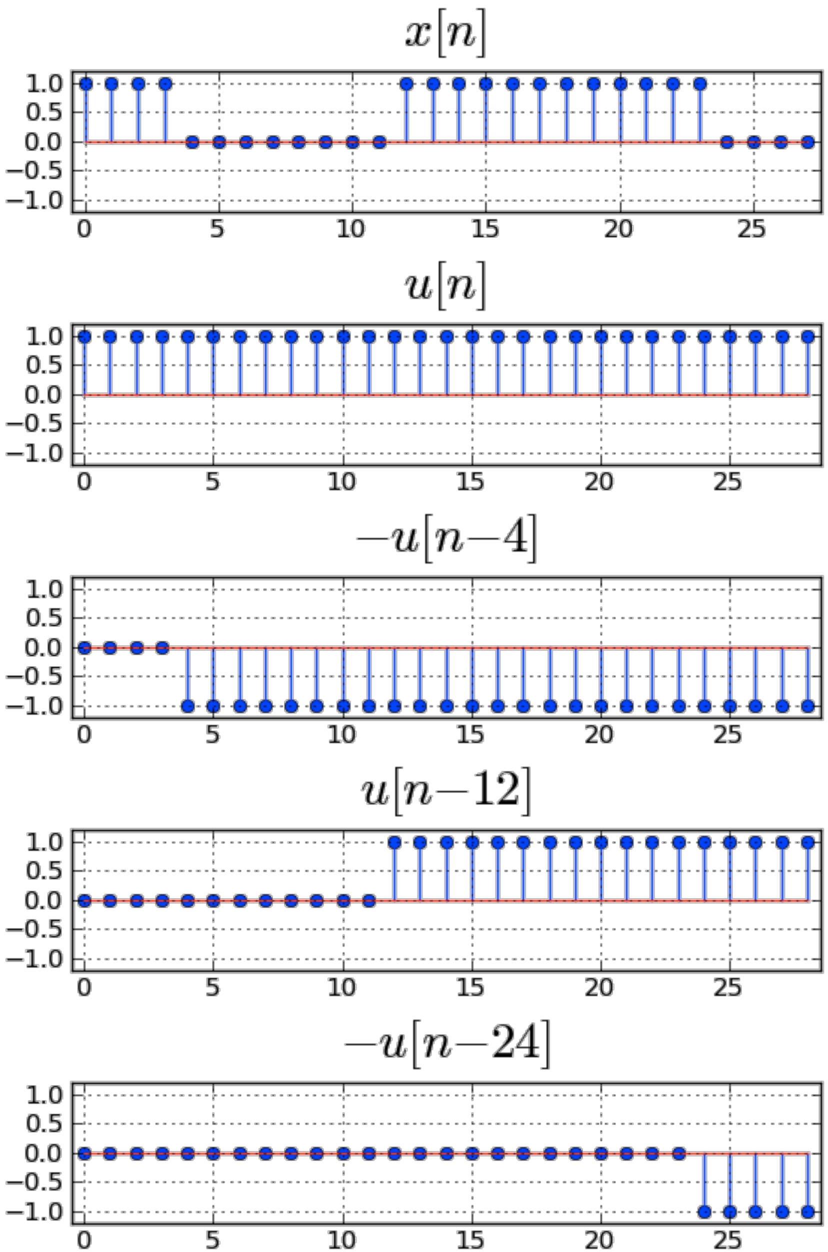
$$\delta[n] = u[n] - u[n-1] \quad \Longrightarrow \quad h[n] = s[n] - s[n-1]$$

from which it follows that

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

(assuming $s[-\infty] = 0$, i.e., a causal LTI system)

Unit Step Decomposition



"Rectangular-wave" digital signaling waveforms, of the sort we have been considering, are easily decomposed into time-shifted, scaled unit steps --- each transition corresponds to another shifted, scaled unit step.

e.g., if x[n] is the transmission of 1001110 using 4 samples/bit:

$$x[n] = u[n] - u[n-4] + u[n-12] - u[n-24]$$

… so the corresponding response is

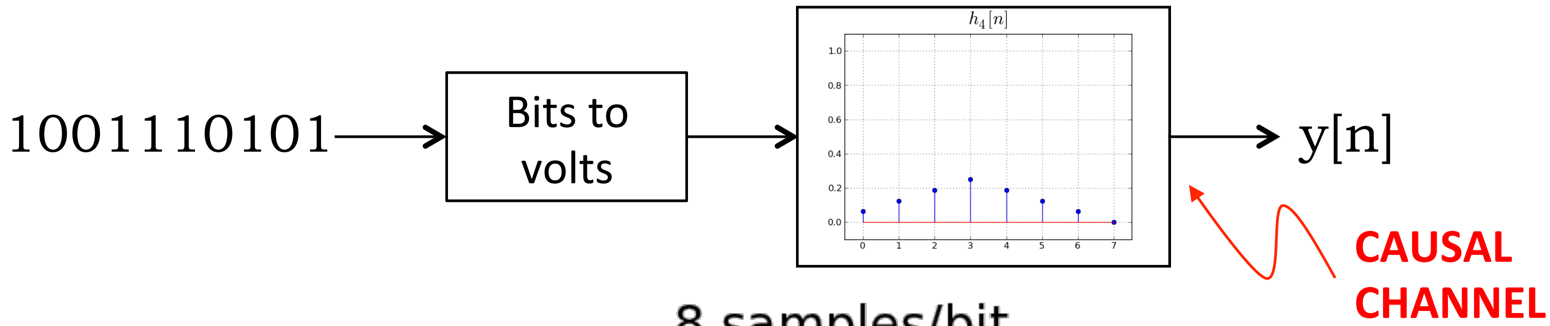
… so the corresponding response is

$$x[n] = u[n] - u[n-4] + u[n-12] - u[n-24]$$

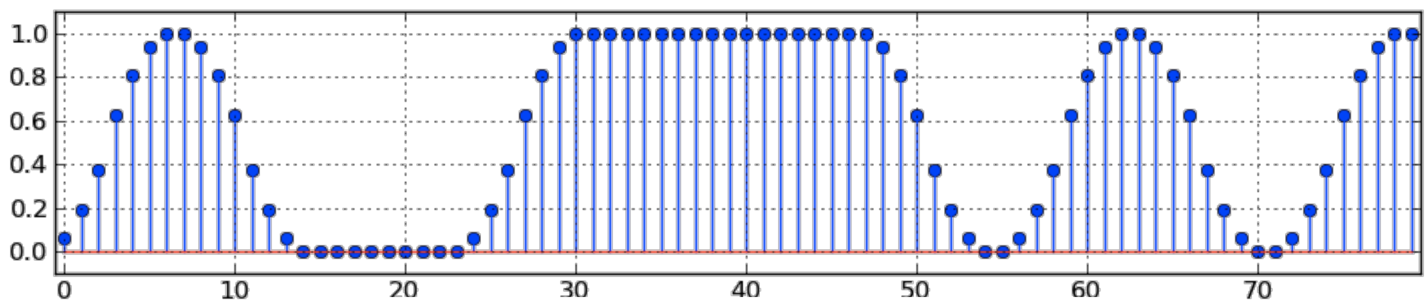
$$y[n] = s[n] - s[n-4] + s[n-12] - s[n-24]$$

Note how we have invoked linearity and time invariance!

Transmission Over a Channel



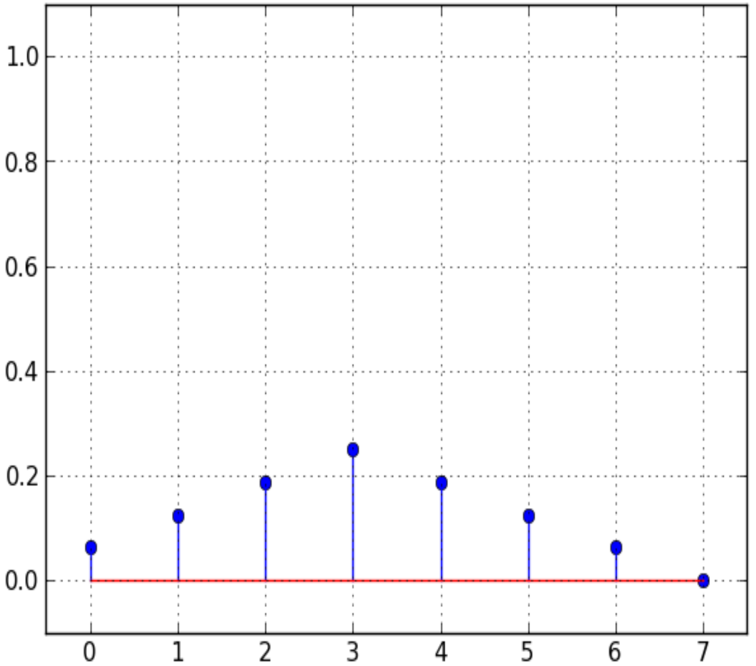
8 samples/bit



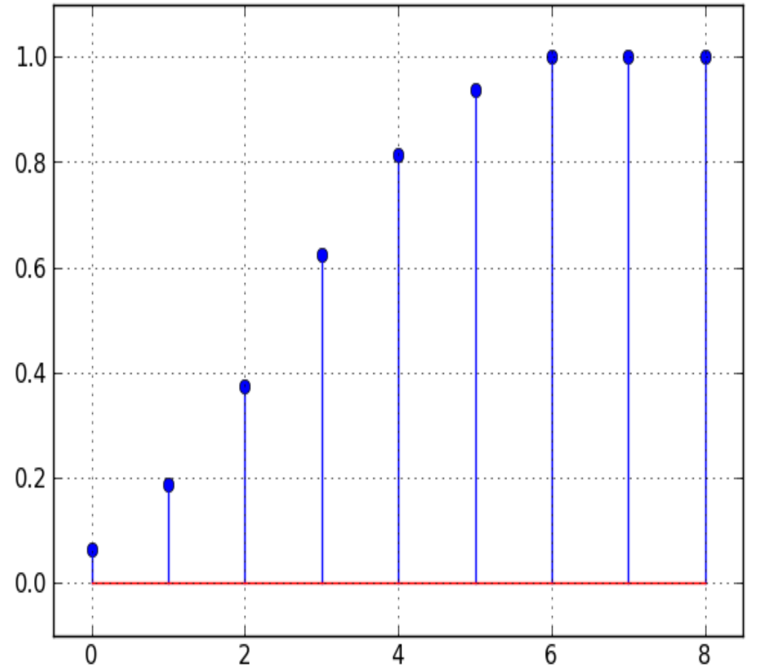
Response of Channel

Example of unit sample response h[n] and corresponding unit step response s[n] for a causal channel model:

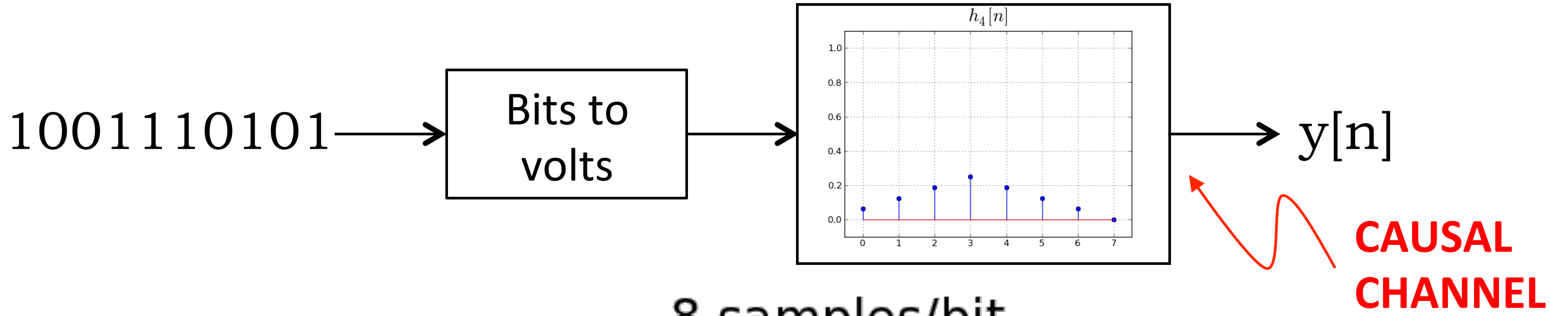
h[n]



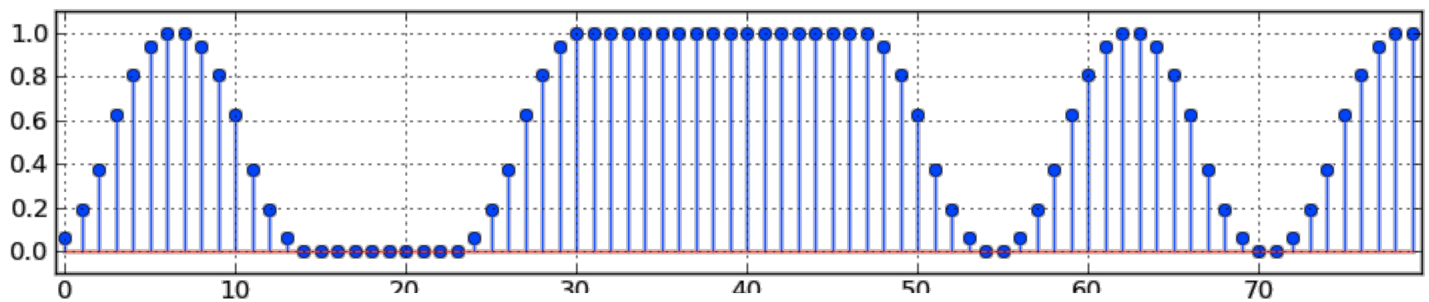
s[n]



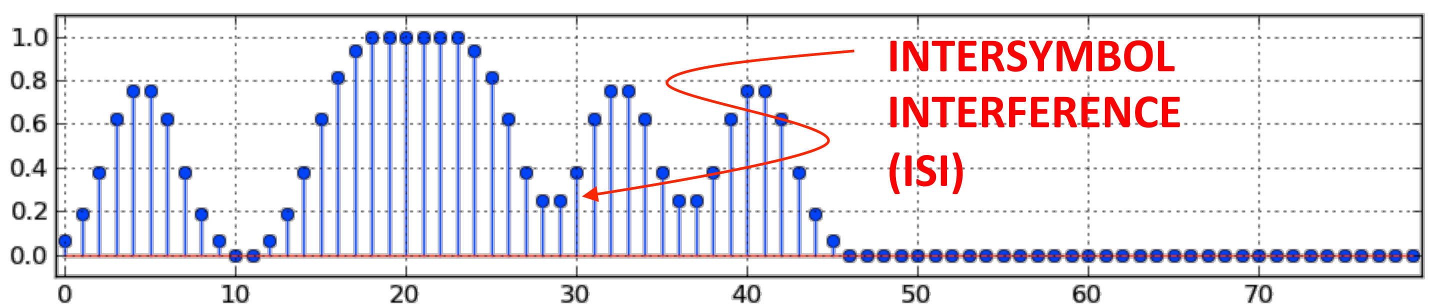
Transmission Over a Channel



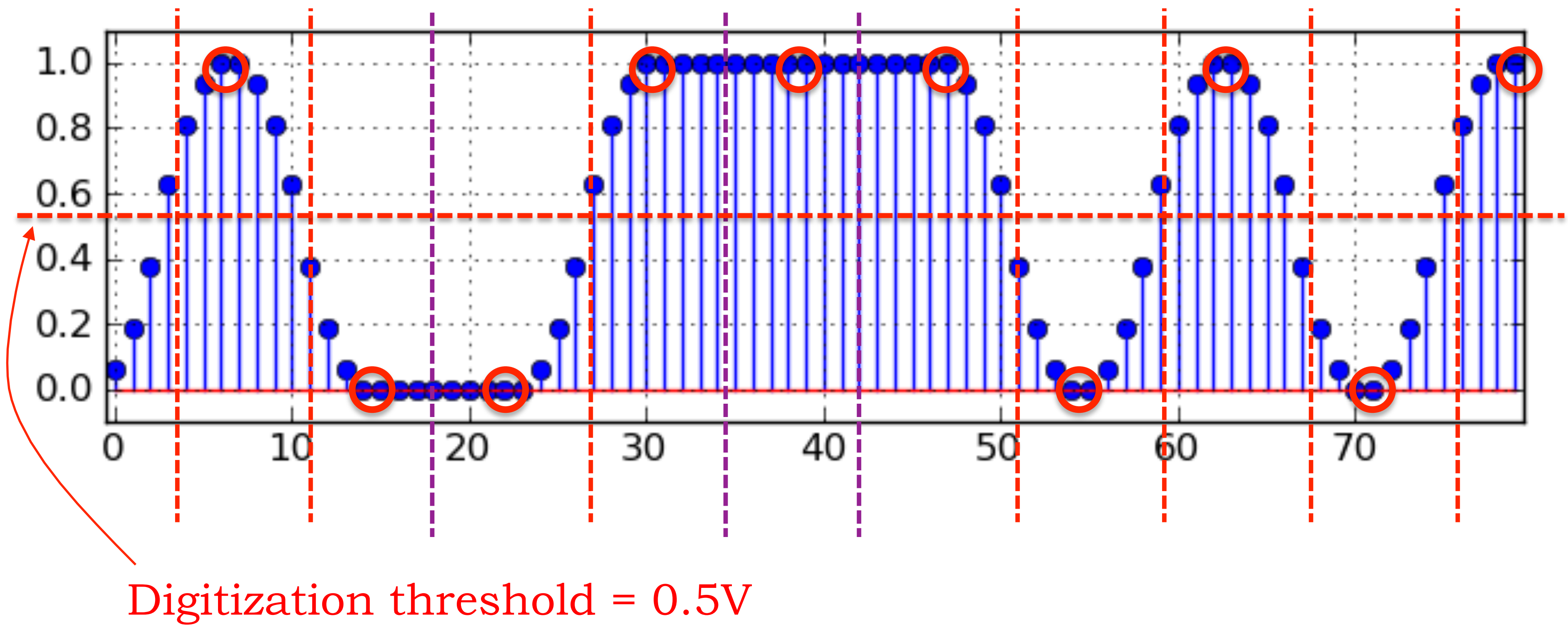
8 samples/bit



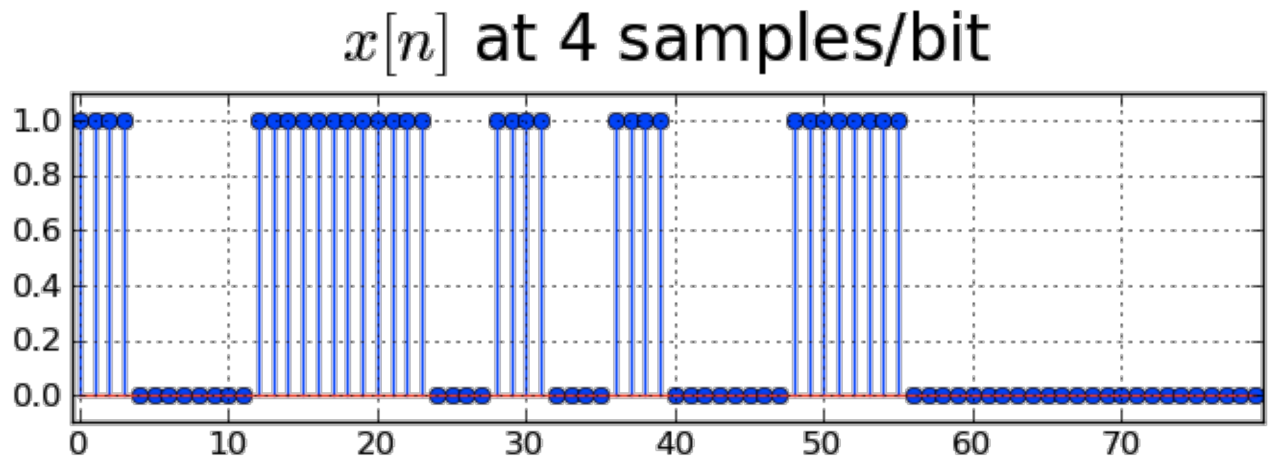
4 samples/bit

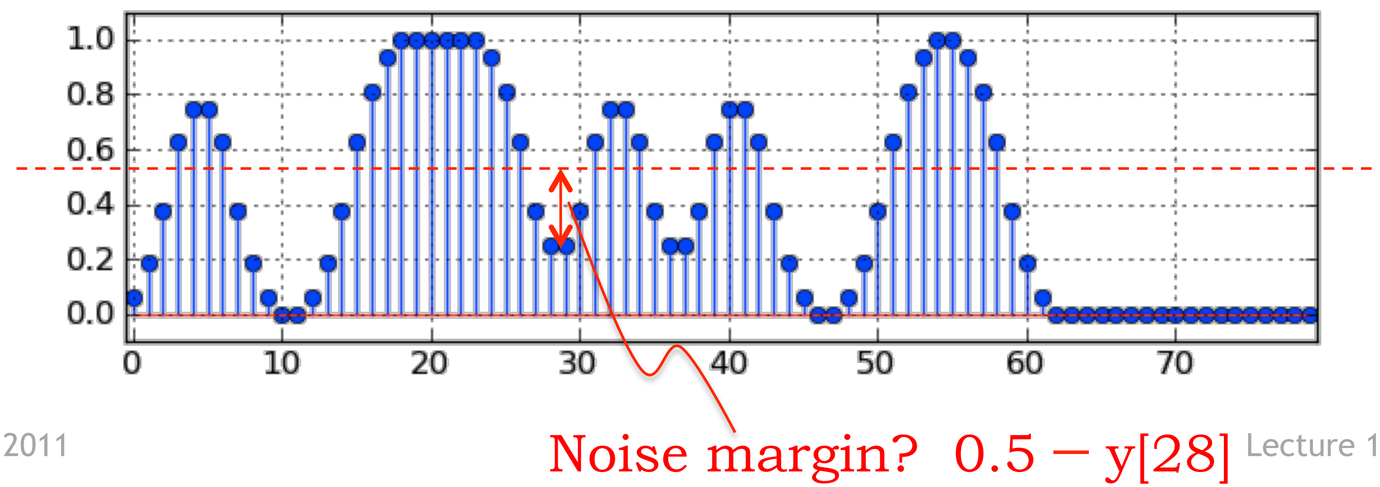


Receiving the Response



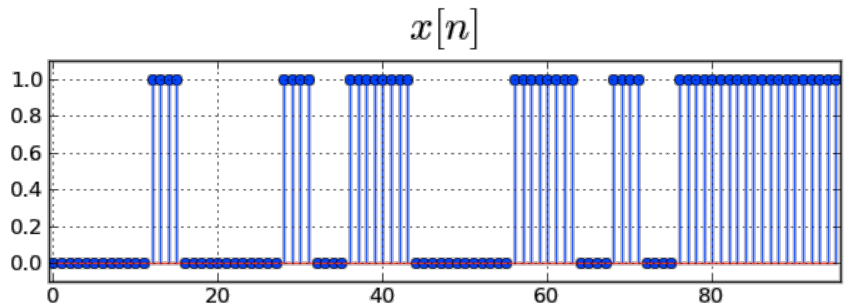
Faster Transmission



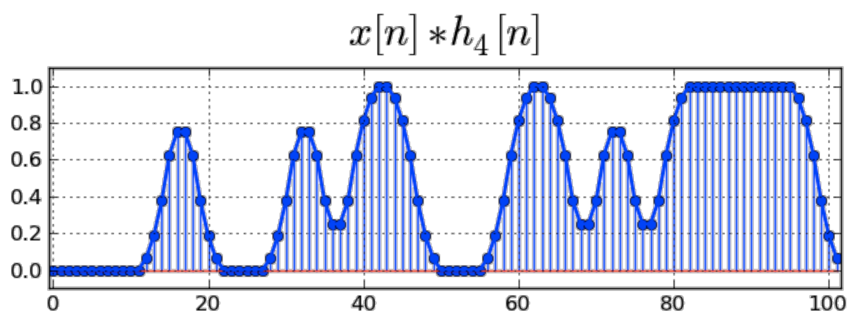


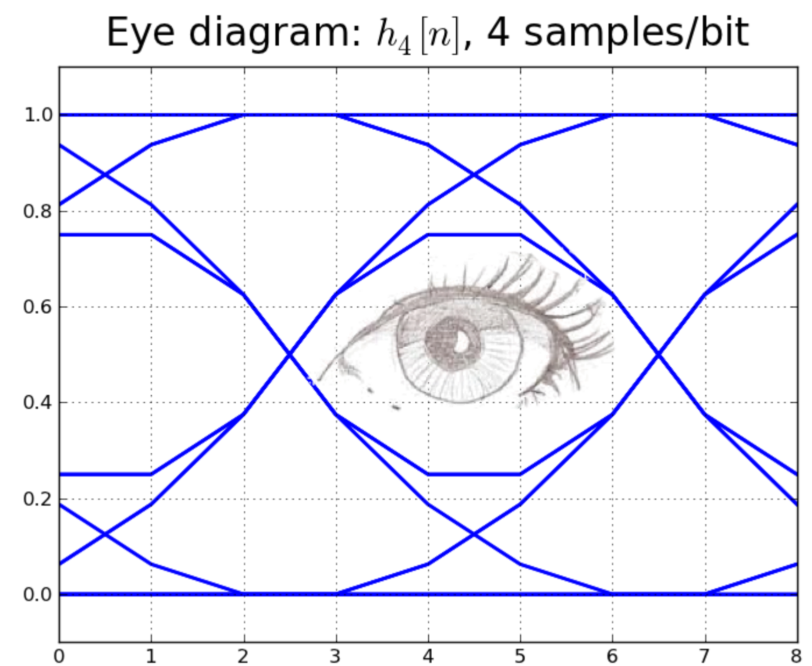
Eye Diagrams

Using same h[n] as before and samples_per_bit=4



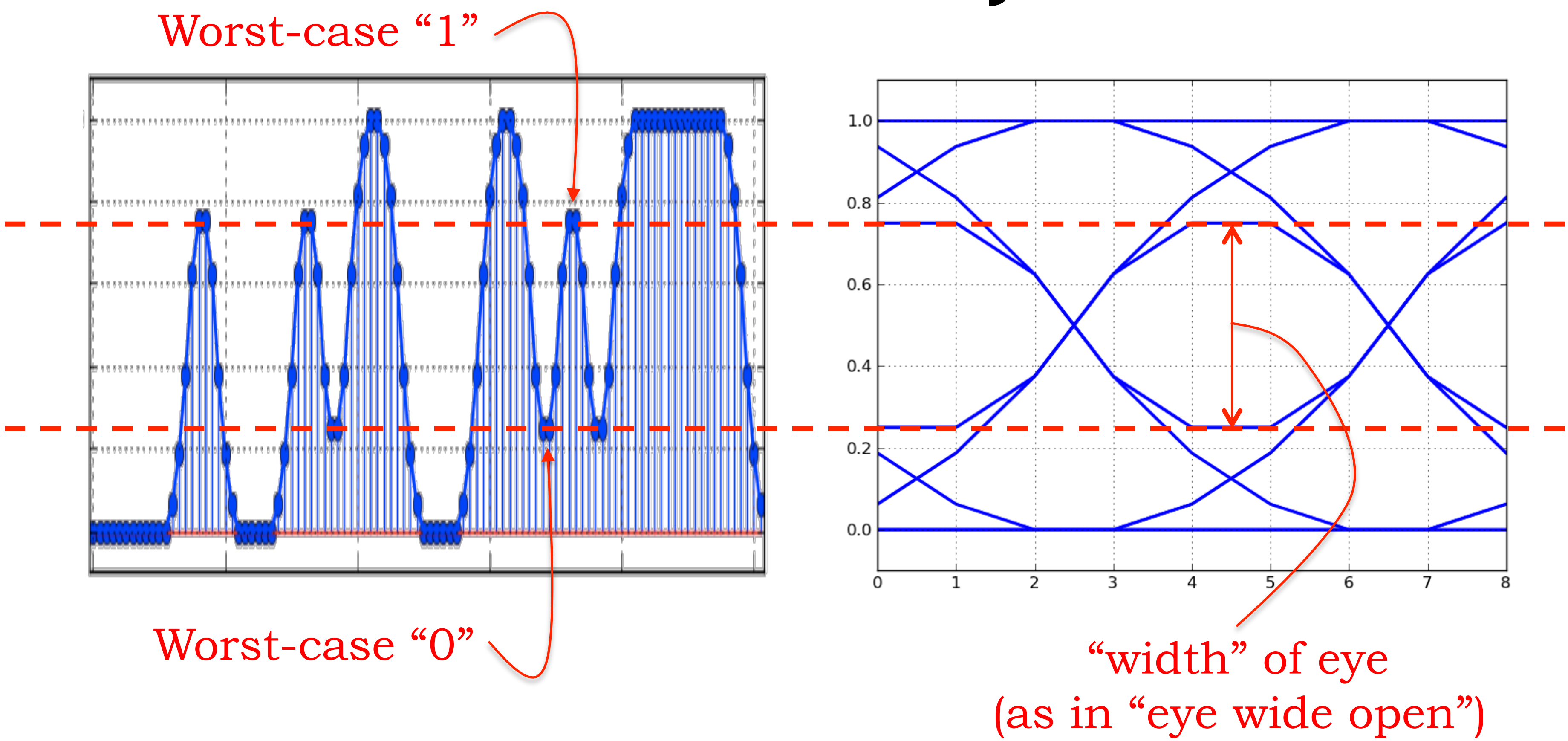
000 100 010 110 001 101 011 111





Eye diagrams make it easy to find the worst-case signaling conditions at the receiving end.

“Width” of Eye



To maximize noise margins:

Pick the best sample point → widest point in the eye

Pick the best digitization threshold → half-way across width

Constructing the Eye Diagram

1. Compute B, the number bits "covered" by h[n]. Let N = samples/bit

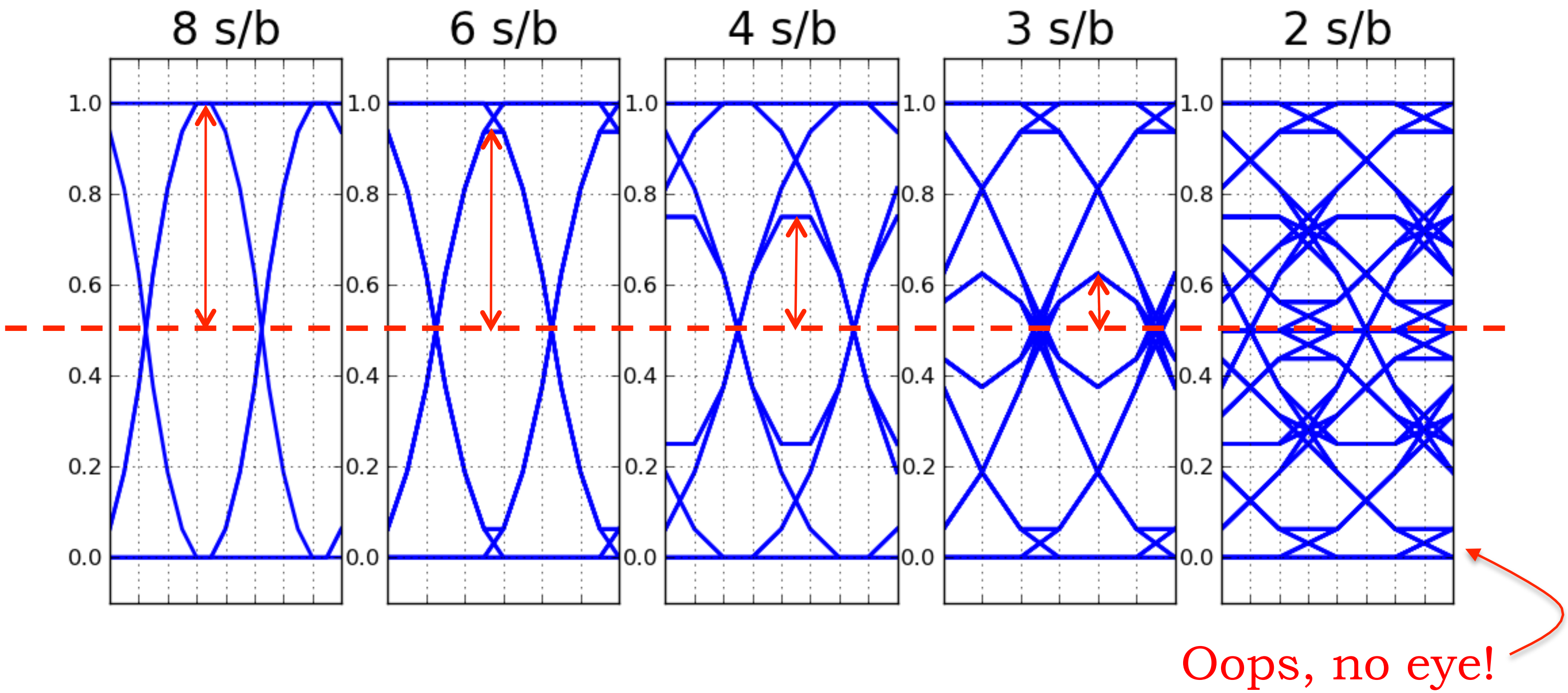
$$B = \left\lfloor \frac{\text{length of active portion of h[n]}}{\text{N}} \right\rfloor + 2$$

2. Generate a test pattern that contains all possible combinations of B bits – want all possible combinations of neighboring cells. If B is big, randomly choose a large number of combinations.

3. Transmit the test pattern over the channel (2^BBN samples)

4. Instead of one long plot of y[n], plot the response as an *eye diagram:*
   a. break the plot up into short segments, each containing KN samples, starting at sample 0, KN, 2KN, 3KN, … (e.g., K=3)
   b. plot all the short segments on top of each other

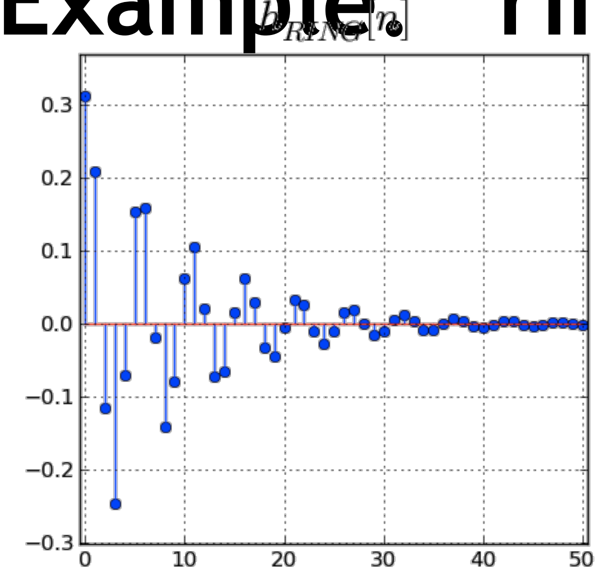
Choosing Samples/Bit

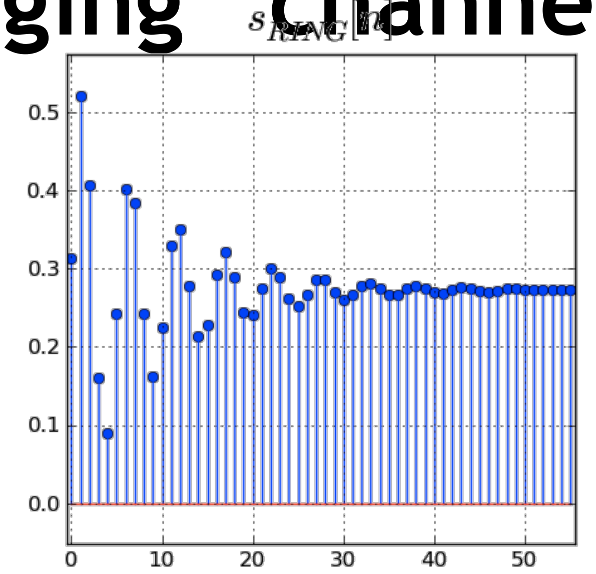


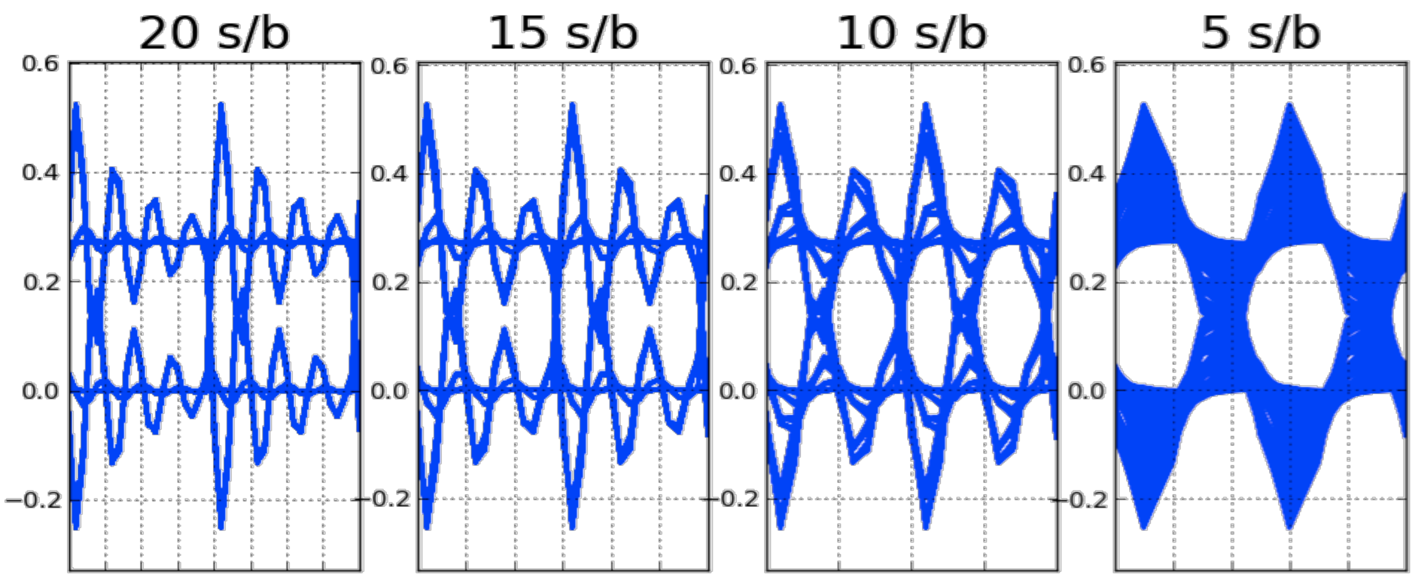
Given h[n], you can use the eye diagram to pick the number of samples transmitted for each bit (N):

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

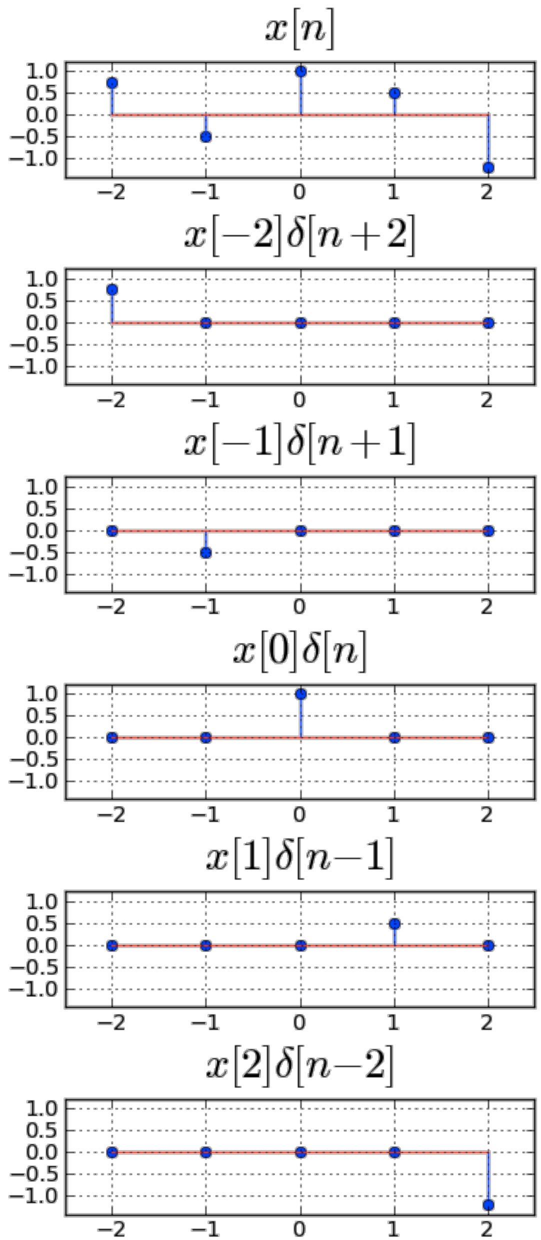
Example: "ringing" channel







From Unit Step Decomposition to Unit Sample Decomposition …



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

$x[-2]\delta[n+2] + x[-1]\delta[n+1] + \ldots + x[2]\delta[n-2]$.

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 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

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

CONVOLUTION SUM

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

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

Notation, Notation!

Evaluating the convolution sum

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

for all n defines the output signal y in terms of the input x and unit-sample response h. Some constraints are needed to ensure this infinite sum is well behaved, i.e., doesn't "blow up" (we'll discuss this soon).

We use $*$ to denote convolution, and write y=x∗h. We can then write the value of y at time n, which is given by the above sum, as $y[n] = (x*h)[n]$.

Notation, Notation!

Evaluating the convolution sum

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

for all n defines the output signal y in terms of the input x and unit-sample response h. Some constraints are needed to ensure this infinite sum is well behaved, i.e., doesn't "blow up" (we'll discuss this soon).

We use $*$ to denote convolution, and write y=x∗h. We can thus write the value of y at time n, which is given by the above sum, as $y[n]=(x*h)[n]$

Be warned: you'll find people writing $y[n]=x[n]*h[n]$, where the poor index n is doing triple duty. This is **awful** notation, but a super-majority of engineering professors (including at MIT) will inflict it on their students.

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.

These two observations allow us to rework the convolution sum when it's used to describe transmission channels:

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

start at t=0 causal j=n-k

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:

$$h_2*(h_1*x) = \left(h_2*h_1\right)*x$$

Convolution is distributive:

$$\left(h_1+h_2\right)*x = (h_1*x)+(h_2*x)$$

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.,

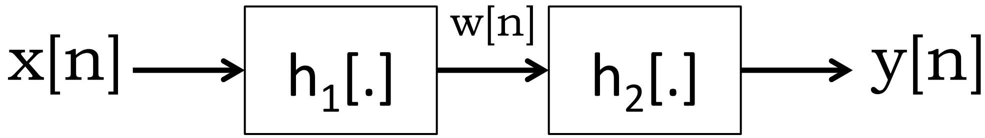
$$\sum_{m=-\infty}^{\infty} |h[m]| < \infty$$

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

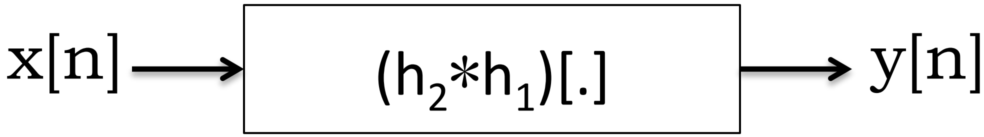
Under these conditions, the convolution sum is well-behaved, and the *output* is guaranteed to be *bounded*.

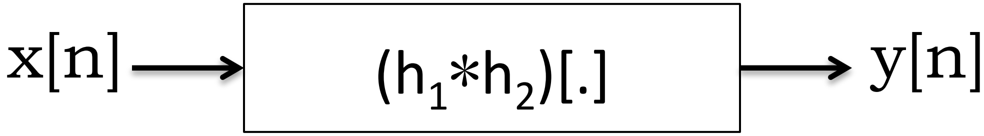
The absolute summability of h[n] is necessary and sufficient for this bounded-input bounded-output (BIBO) stability.

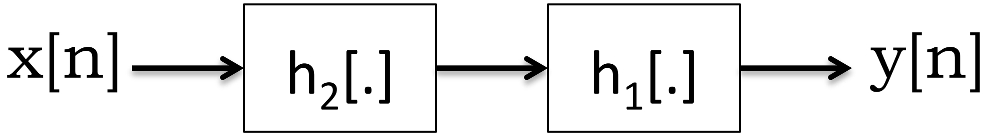
Series Interconnection of LTI Systems



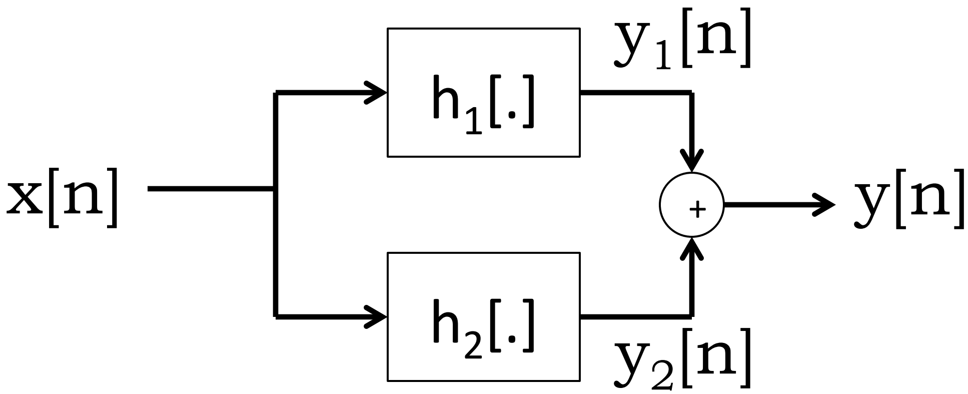
$$y = h_2 * w = h_2 * \left(h_1 * x\right) = \left(h_2 * h_1\right) * x$$







Parallel Interconnection of LTI Systems



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