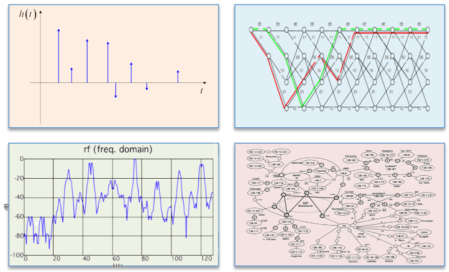


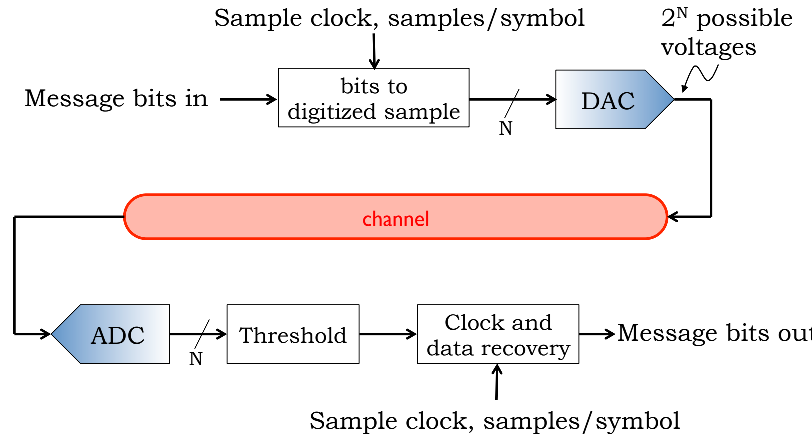
INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

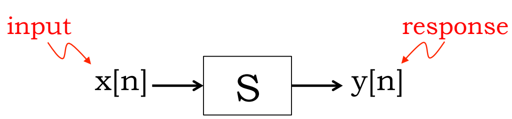
6.02 Spring 2011
Lecture #4

- Inputs & responses
- Linear time-invariant systems
- Modeling communications channels

Today: Modeling Channel Behavior



System Input and Response



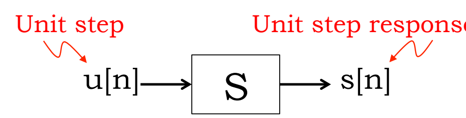
A discrete-time signal is described by an infinite sequence of values, denoted by x[n], y[n], z[n], and so on. The indices fall in the range $-\infty$ to $+\infty$.

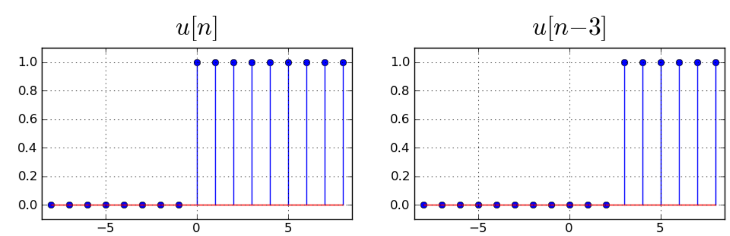
In the diagram above, the sequence of output values y[n] is called the *response* of system S to the *input* sequence x[n].

Unit Step and Unit Step Response

A simple but useful discrete-time signal is the *unit step,* u[n], defined as

$$u[n] = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$$

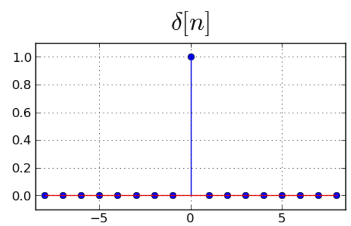


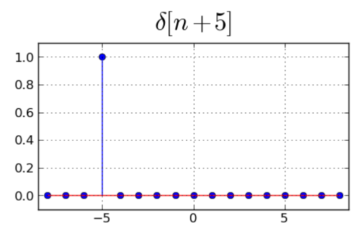


Unit Sample

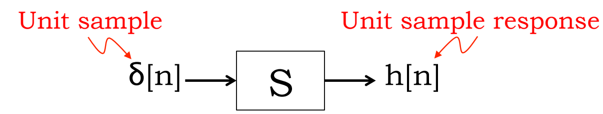
Another simple but useful discrete-time signal is the *unit sample,* δ[n], defined as

$$\delta[n] = u[n] - u[n-1] = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases}$$



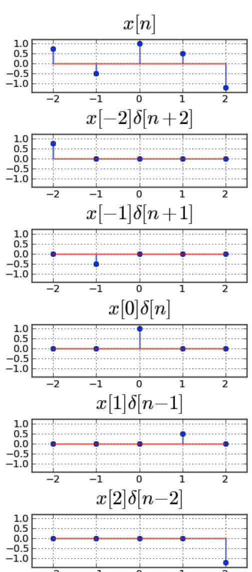


Unit Sample Response



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

Unit-sample Decomposition



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

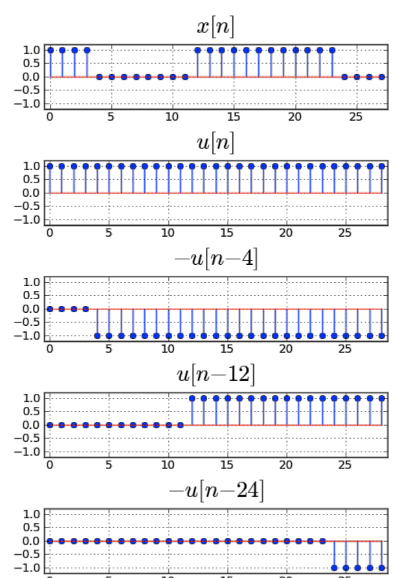
Example: in the figure, x[n] is the sum of x[-2]δ[n+2] + x[-1]δ[n+1] + … + x[2]δ[n-2].

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

Unit-step Decomposition



Digital signaling waveforms are easily decomposed into time-shifted, scaled unit steps (each transition corresponds to another shifted, scaled unit step).

In this example, x[n] is the transmission of 1001110 using 4 samples/bit:

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

Time Invariant Systems

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

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

$$x[n-N] \longrightarrow \boxed{S} \longrightarrow y[n-N]$$

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

Linear Systems

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

If

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

then system S is said to be *linear*. If the input is the weighted sum of several signals, the response is the *superposition* (i.e., weighted sum) of the response to those signals.

Modeling LTI Systems

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 samples

Sum of shifted, scaled responses

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

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

$$x[n] \longrightarrow \boxed{h_S[n]} \longrightarrow y[n]$$

Properties of Convolution

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

The summation is called the convolution sum, or more simply, the *convolution* of x[n] and h[n]. "*" is the convolution operator.

Convolution is commutative:

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

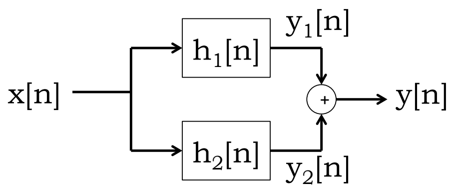
Convolution is associative:

$$x[n]*\left(h_1[n]*h_2[n]\right)=\left(x[n]*h_1[n]\right)*h_2[n]$$

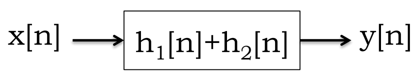
Convolution is distributive:

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

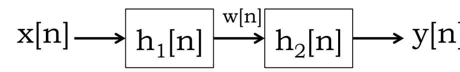
Parallel Interconnection of LTI Systems



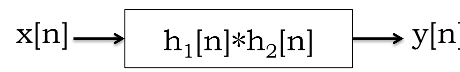
$$y[n] = y_1[n] + y_2[n] = x[n]*h_1[n] + x[n]*h_2[n] = x[n]*\left(h_1[n] + h_2[n]\right)$$

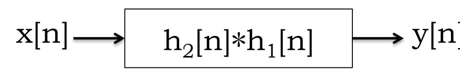


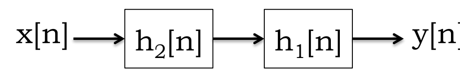
Series Interconnection of LTI Systems



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







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

Relationship between h[n] and s[n]

We're often given one of h[n] or s[n] and would like to know the other. On slide #5 we saw

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

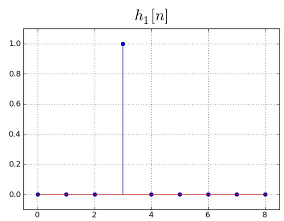
Which for LTI systems implies

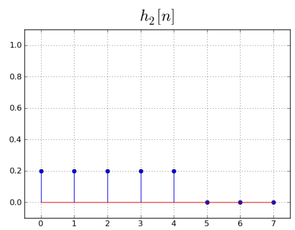
$$h[n] = s[n] - s[n-1]$$

In other words, the unit sample response is the first difference of the unit step response. Also

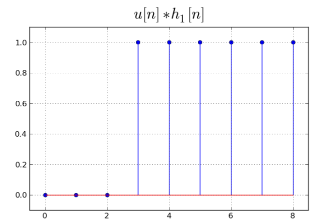
$$s[n] = \sum_{k=-\infty}^{n} h[k]$$

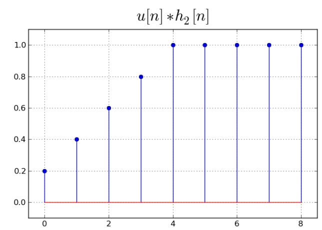
h[n]



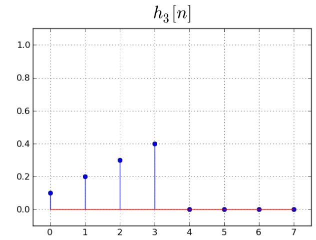


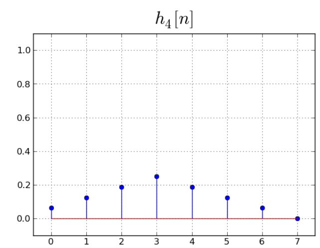
s[n]=u[n]*h[n]



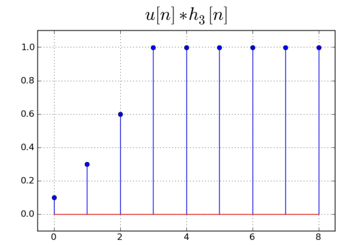


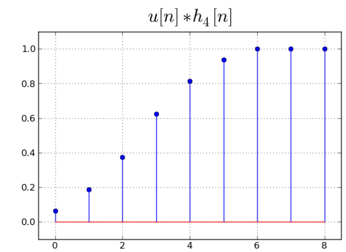
h[n]



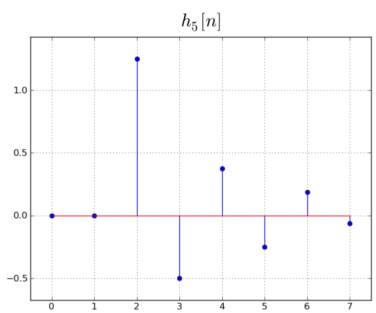


s[n]=u[n]*h[n]

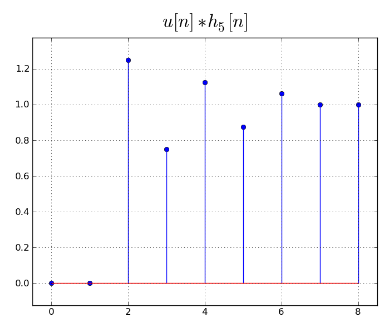




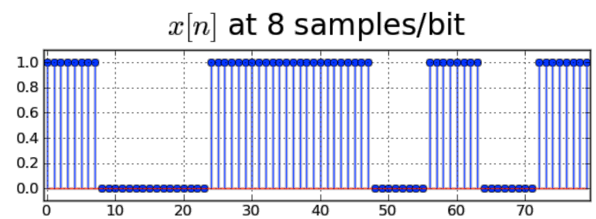
h[n]

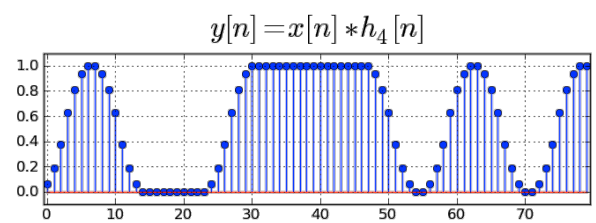


s[n]=u[n]*h[n]

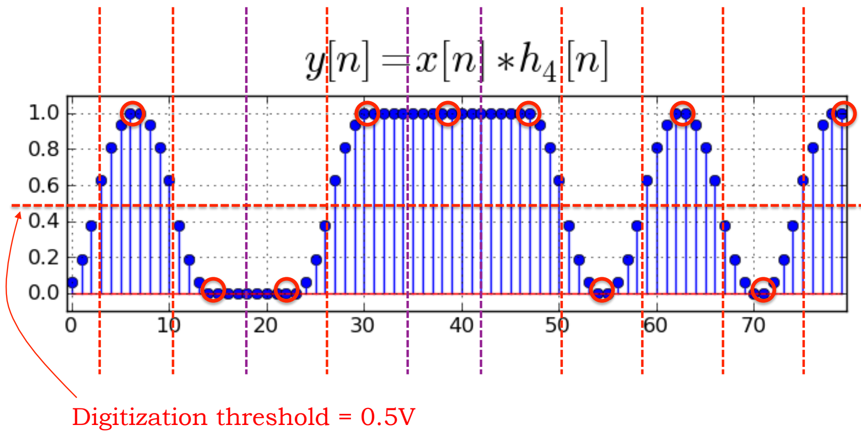


Transmission Over a Channel

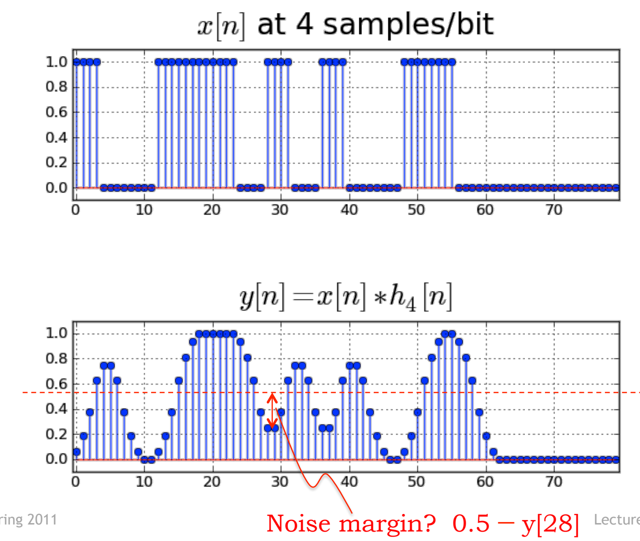




Receiving the Response



Faster Transmission



Computing y[28] using $s_4[n]$

n	$s_4[n]$
<0	0.0
0	0.0625
1	0.1875
2	0.375
3	0.625
4	0.8125
5	0.9375
≥6	1.0

We can use $s_4[n]$ to compute y[28] as follows:

$$x[n] = u[n] - u[n-4] + u[n-12] - u[n-24] + u[n-28] + \ldots$$

So

$$y[n] = s_4[n] - s_4[n-4] + s_4[n-12] - s_4[n-24] + s_4[n-28] + \ldots$$

For n=28

$$\begin{aligned} y[28] &= s_4[28] - s_4[28-4] + s_4[28-12] - s_4[28-24] + s_4[28-28] + \ldots \\ &= s_4[28] - s_4[24] + s_4[16] - s_4[4] + s_4[0] + \ldots \\ &= 1.0 - 1.0 + 1.0 - 0.8125 + 0.0625 \\ &= 0.25 \end{aligned}$$

So the noise margin is 0.5-0.25 = 0.25V.

Computing y[28] using $h_4[n]$

n	$h_4[n]$
<0	0.0
0	0.0625
1	0.125
2	0.1875
3	0.25
4	0.1876
5	0.125
6	0.0625
≥7	0.0

We can use $h_4[n]$ to compute y[28] as follows: first expand convolution sum keeping non-zero $h_4[n]$ terms (see bottom right, slide #15):

$$\begin{aligned} y[n] = {} & x[n]h_4[0] + x[n-1]h_4[1] + x[n-2]h_4[2] + x[n-3]h_4[3] + \\ & x[n-4]h_4[4] + x[n-5]h_4[5] + x[n-6]h_4[6] \end{aligned}$$

For n=28:

$$\begin{aligned} y[28] = {} & x[28]h_4[0] + x[27]h_4[1] + x[26]h_4[2] + x[25]h_4[3] + \\ & x[24]h_4[4] + x[23]h_4[5] + x[22]h_4[6] \end{aligned}$$

$$\begin{aligned} &= 0.0625 + 0 + 0 + 0 + 0 + 0.125 + 0.0625 \\ &= 0.25 \end{aligned}$$

This agrees with the previous calculation. ✓