Topic

Filters, Reverberation & Convolution

Lecture by: Fatemeh Pishdadian

Lecture Outline

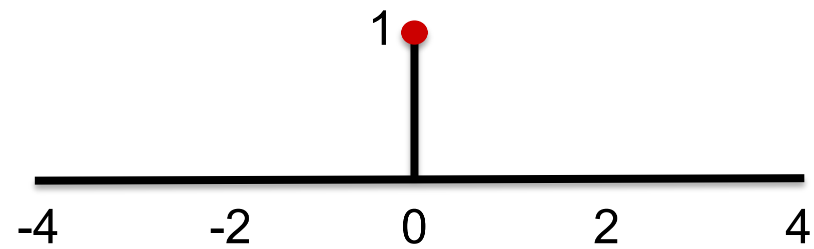
- What's an Impulse?
- LTI systems
- Frequency selective filters
- Delay and moving average
- Reverberation

The impulse function

Impulse response

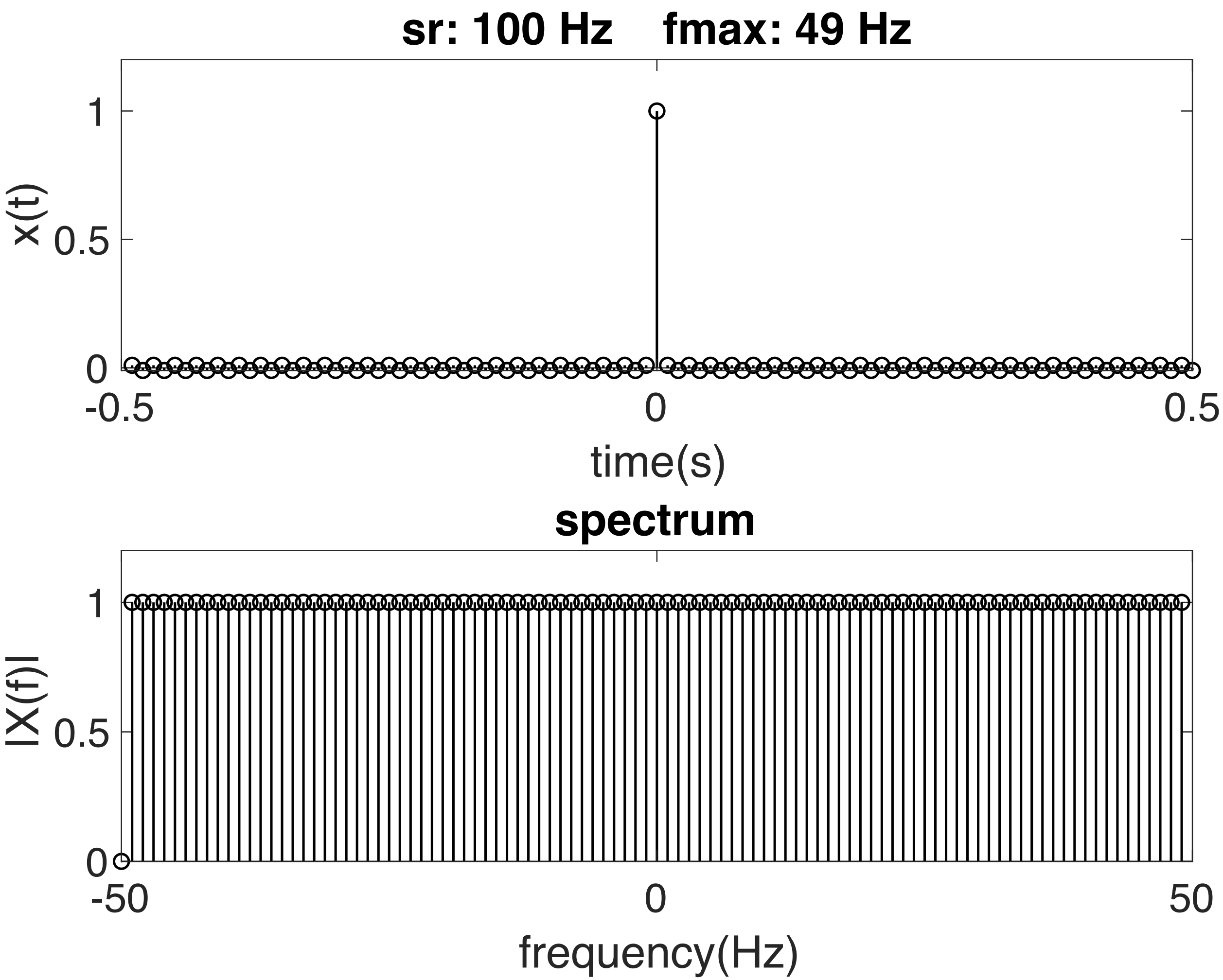
- An "impulse" is this signal:

$$\delta[n] = \left\{ \begin{array}{ll} 1 & \text{if n} = 0 \\ 0 & \text{else} \end{array} \right\}$$



- The next slide shows an impulse (top) and the frequency representation of that impulse you get by plotting the absolute value of the Fourier transform of that impulse.
- Note, the impulse shows energy at ALL frequencies of analysis.

Impulse function



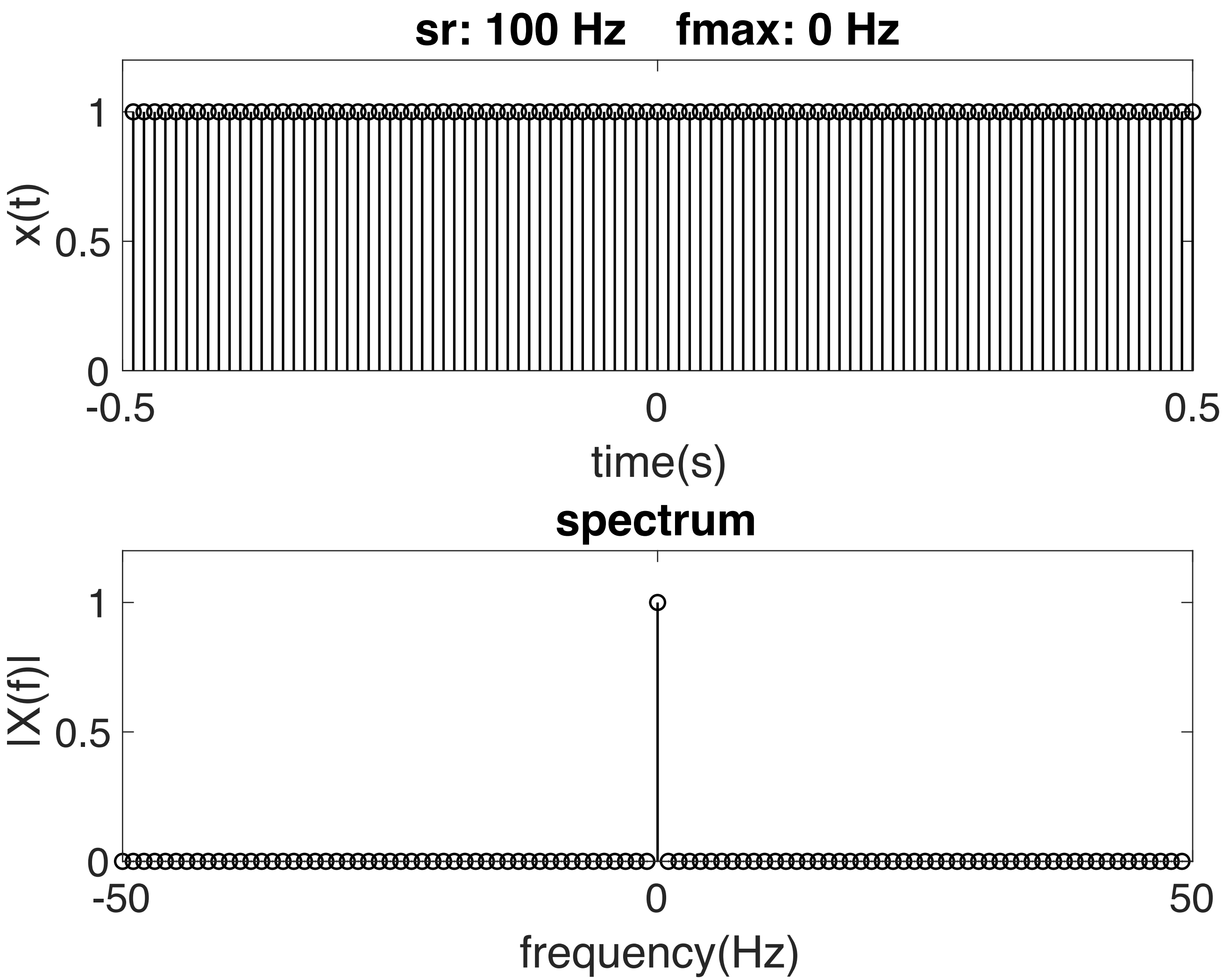
Let’s look at the D.C offset

- Next slide, you’ll see a time-domain signal (top panel) that is all 1. This is a constant signal.

- When you take the FFT and display the magnitude spectrum (bottom panel), you get just one non-zero value, at the 0 frequency. This is the D.C. offset.

- The DC offset is basically the opposite of the impulse function from the previous slide: all times are non-zero, but only one frequency is non-zero.

DC Offset in time and frequency



Making it more broadband

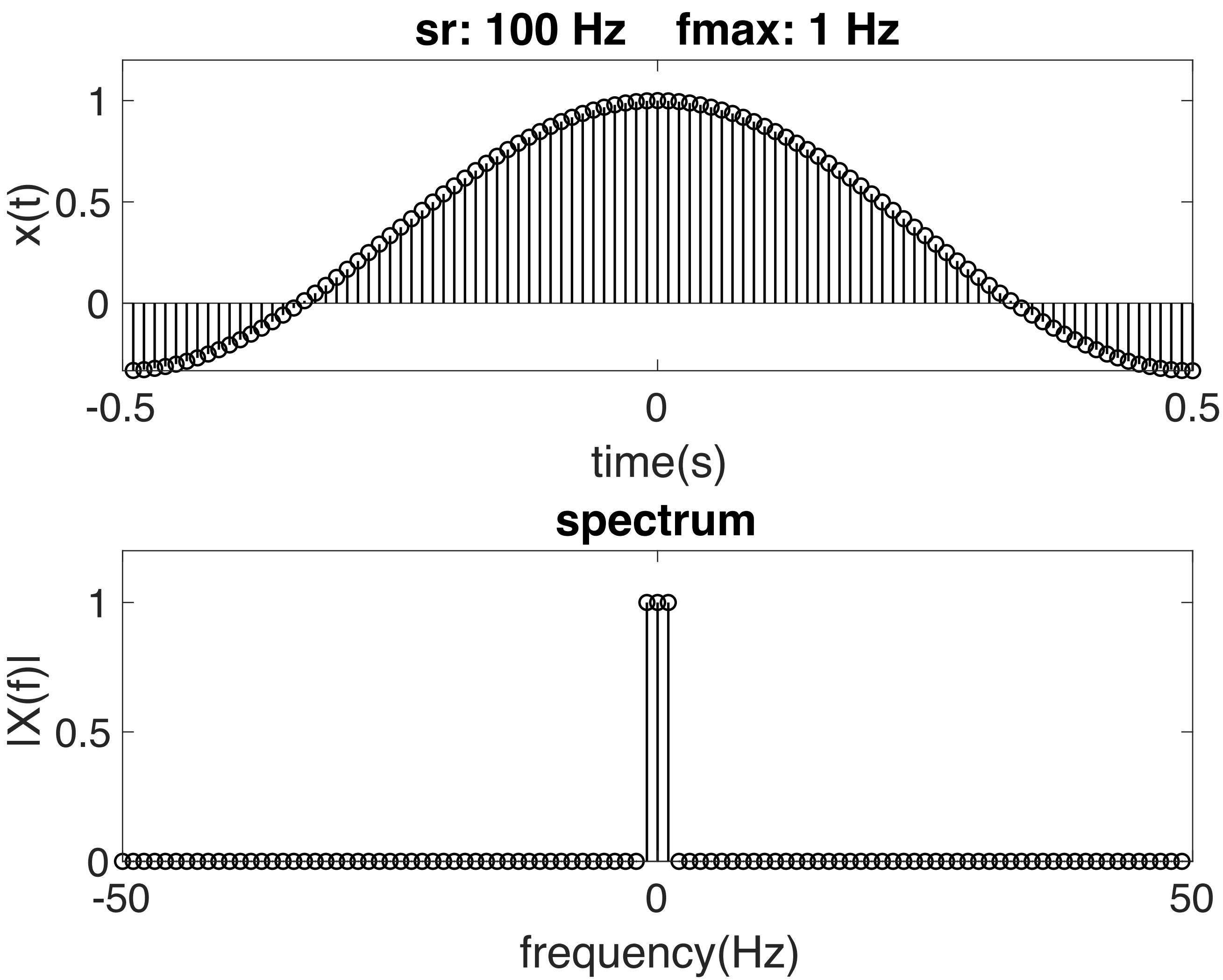
- Now, let's take our DC offset and make it a touch more broadband.

- We'll add energy in the frequency domain X by setting the 1^{st} frequency of analysis (in both the positive and negative frequencies) to 1.

- Then, we do the inverse Fourier transform to see what the signal looks like.

- Note that, this changes what happens in the time domain. Now the energy starts to concentrate towards the middle.

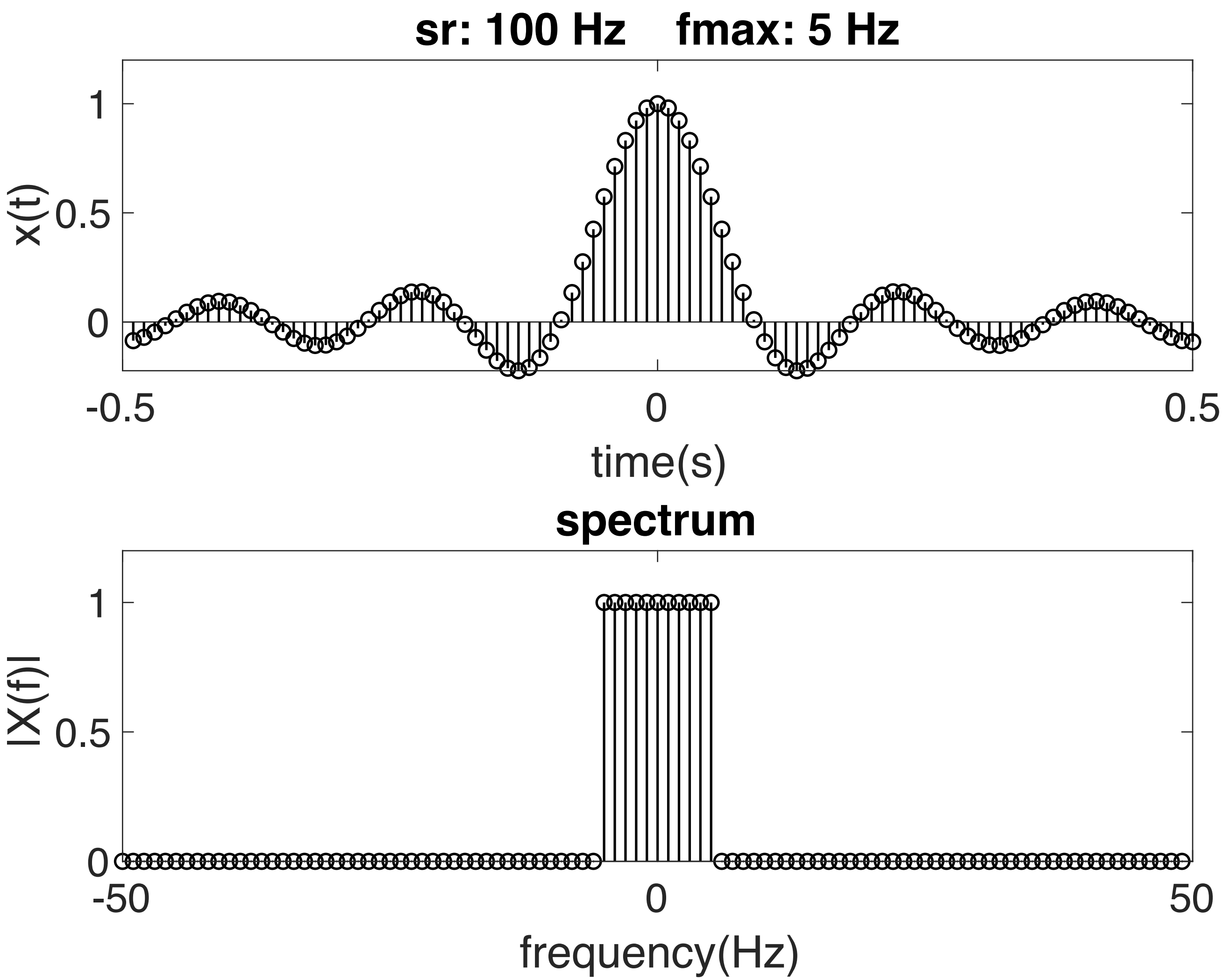
Adding energy at low frequency



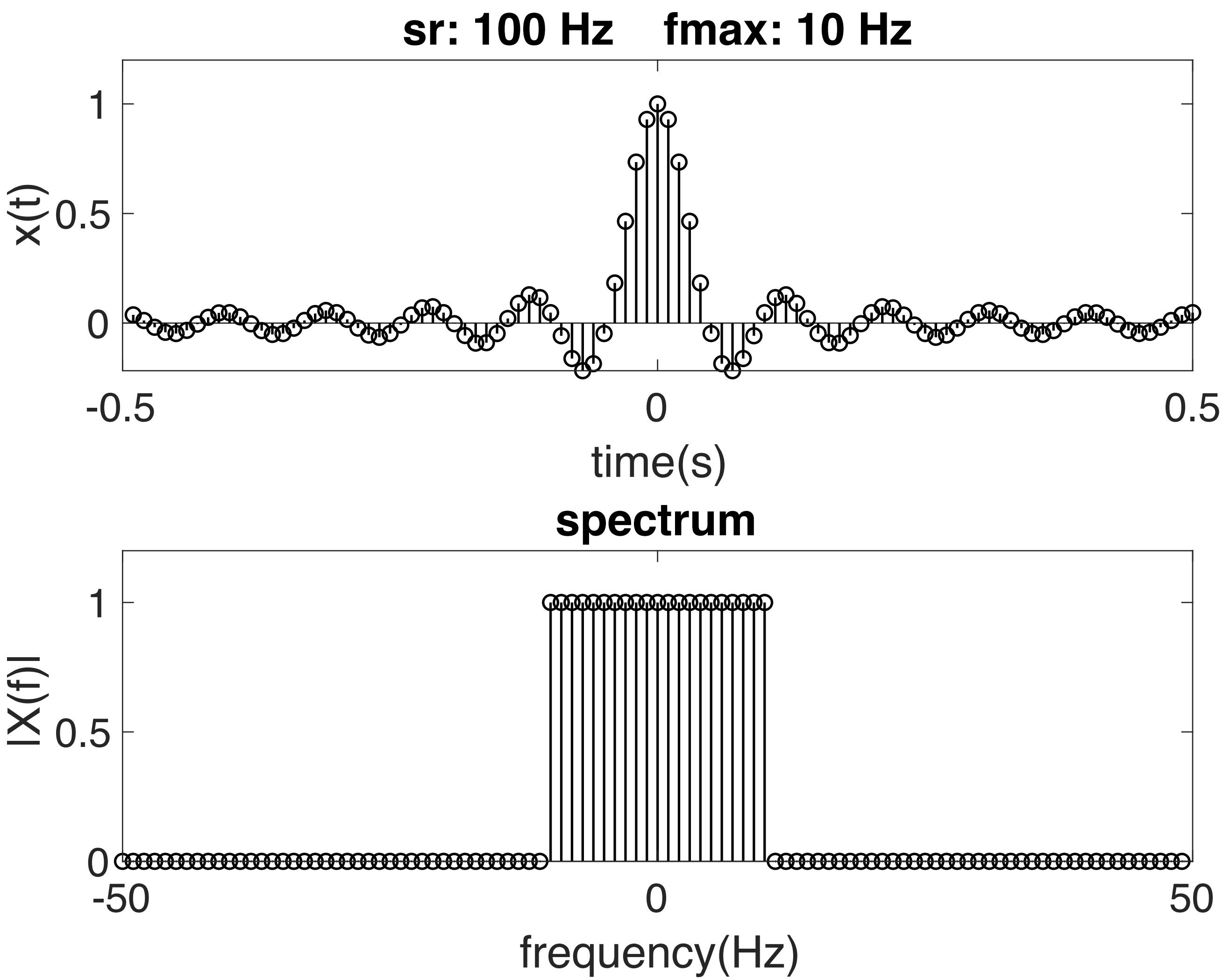
Rinse and repeat

- Over the next few slides, we'll keep adding energy at new frequencies and then take the inverse Fourier transform to see what happens in the time domain.
- As the frequency representation become more broadband (i.e. multiple frequencies have a lot of energy), the time representation of the signal becomes more and more centered on a single point in time.

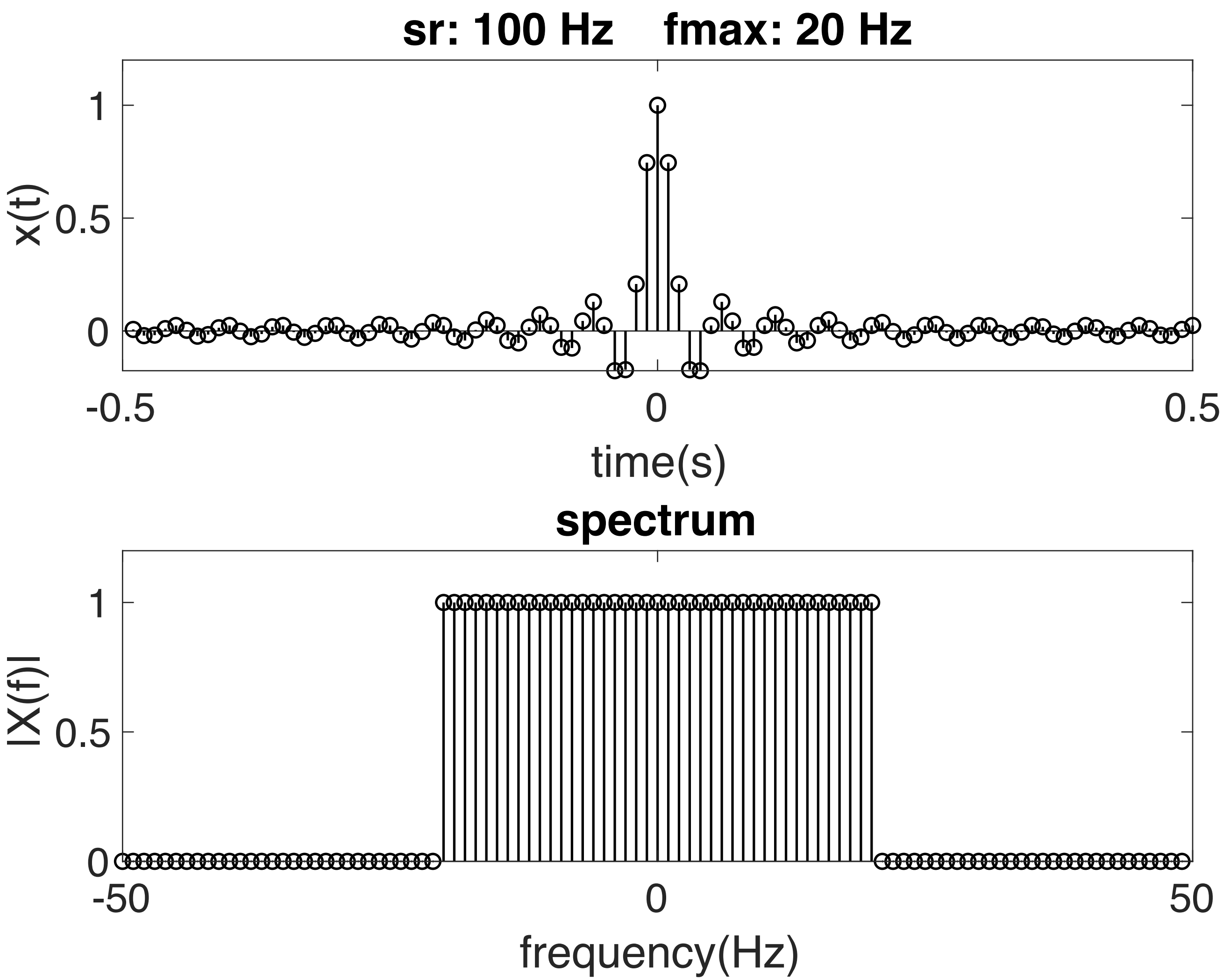
Getting more broadband



Even broader



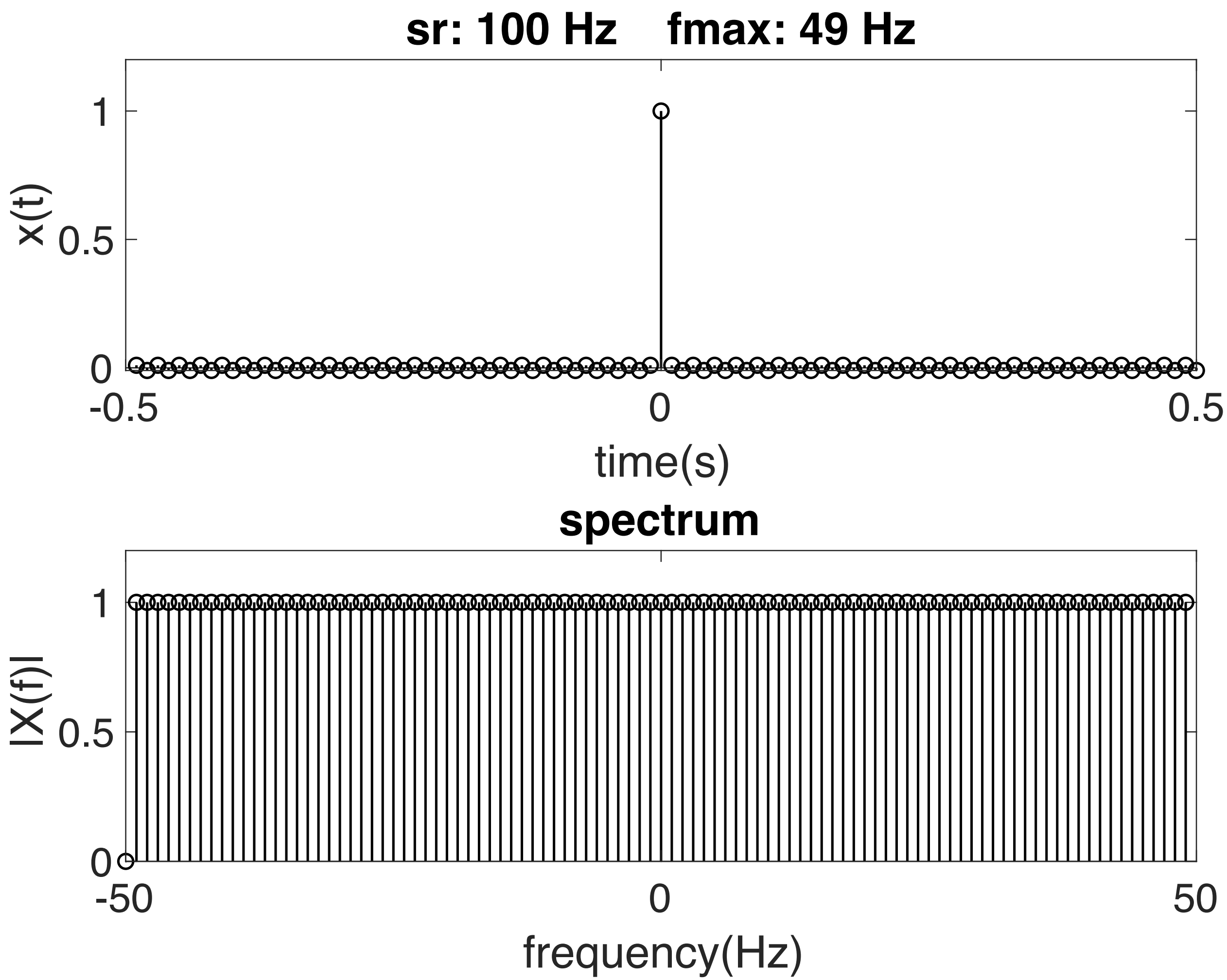
Broader still



How about a signal that has equal energy at all frequencies?

- Next slide, you’ll see a signal that has equal energy at all frequencies.
- As you can see, when we do that, we’re back to the impulse function.

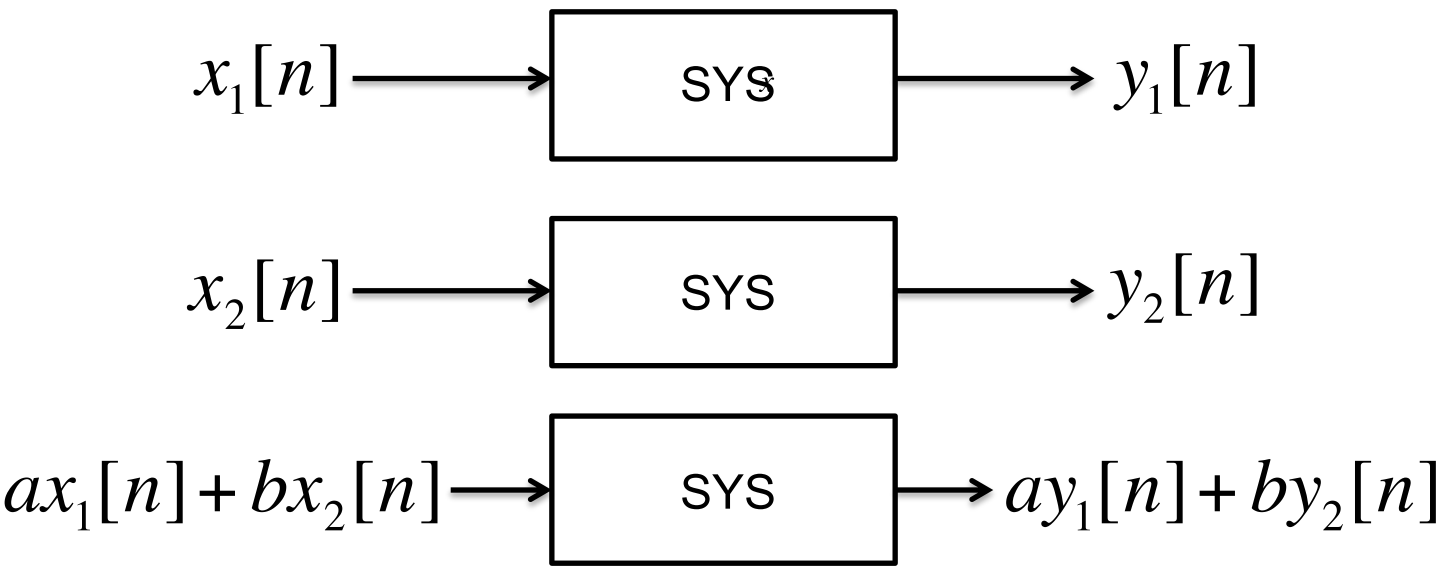
Impulse function



Linear, Time-invariant systems

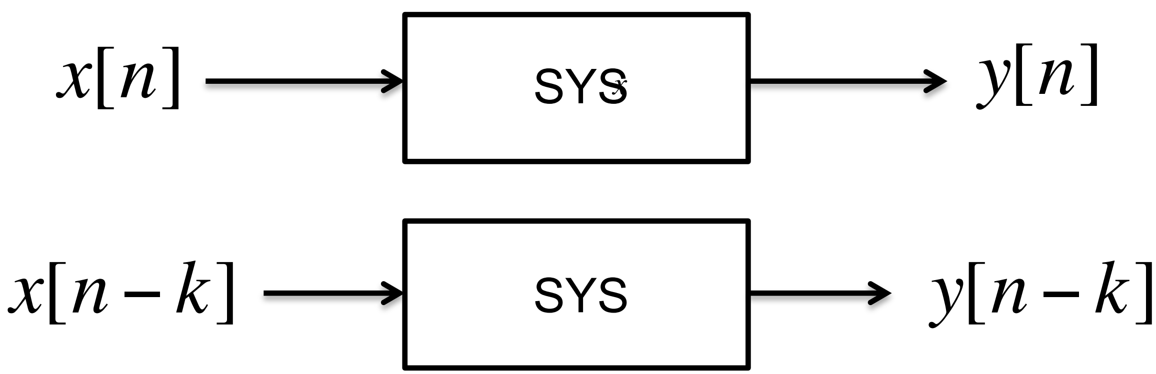
Linearity & Time Invariance

- A system is called linear if it satisfies the superposition property:



Linearity & Time Invariance

- A system is called time invariant if its behavior does not change over time:



LTI systems

- Why do we like to think of systems we work with as LTI?

- Examples of linear systems?

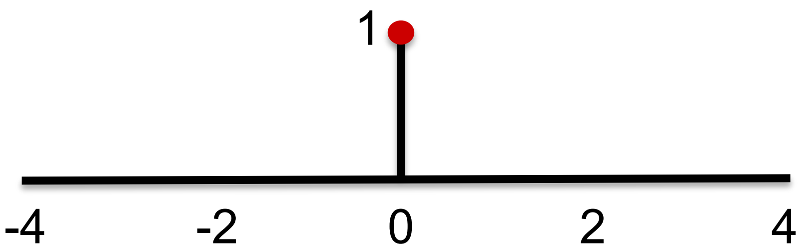
- Examples of time invariant systems?

(We're going to assume rooms, like a classroom or a church, are LTI systems)

Impulse response

- Recall that an “impulse” is this signal:

$$\delta[n] = \left\{ \begin{array}{ll} 1 & \text{if n} = 0 \\ 0 & \text{else} \end{array} \right\}$$



- The ***impulse response h[n]*** of a system is the output of the system when the input is an impulse.

- The ***frequency response H(ω)*** of a system is the Fourier transform of its impulse response h[n].

Impulse response

- An LTI system is fully identified by its impulse response (or frequency response), because…
  - An arbitrary signal x[n] is the sum of scaled and shifted impulse functions:

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

  - Then if we have h[n], by assuming linearity and time invariance we can find the response to x[n]

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

Does this formula look familiar?

Impulse response

- Look! We're back to convolution!

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

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

Does this formula look familiar?

Frequency selective filters

- Frequency selective filters we use in this course are a subset of LTI system.

- The output of filters will be computed via convolution in the time domain, or equivalently, via multiplication in the frequency domain.

Filtering, practically speaking

- I can filter a signal **x[n]** with filter **h[n]** get a filtered signal y[n] by doing convolution between **x[n]** and **g[n]**
- Time-domain convolution takes O(n^2) time. Too long!
- The circular convolution theorem says this…

$$y = h * x = ifft(H \cdot X)$$

- Therefore, we take the fft of **h** to get **H** and the fft of **x** to get **X**.
- Then we element-wise multiply **X** by **H**
- Then we do the inverse fft on the result to get **y**

One caveat

- Let’s say both **x** and **h** are **n** samples long.
- According to the definition of convolution, you should end up with something that is **2n-1** samples long (Go back and look at the python code for of convolution in the course slides)
- For this to happen in frequency, you have to zero-pad both **x** and **h** to be length **2n-1** when you do the Fourier transform.
- This is easy to do as a parameter you pass the fft function.

FIR Filter

- FIR means “Finite Impulse Response”
- Means it will stop making noise once you stop putting noise through it
- There are also Infinite Impulse Response (IIR) filters. These have feedback
- To find out more about IIR filters, take a DSP class.

Building a low-pass FIR filter

- Pick width of your LOW pass band

  (0 Hz to ω_c Hz, where ω_c can be up to the Nyquist rate)

- Create a desired frequency response (don't forget the mirror frequencies above the Nyquist rate).

- Take the IFFT of the frequency response

- This is your impulse response function.

An 8-point 250 Hz low-pass filter

Python Index	0	1	2	3	4	5	6	7
h[n]	0.71	0.26	-0.18	0.06	0.06	0.06	-0.18	0.26
H [k]	1	1	1	0	0	0	1	1

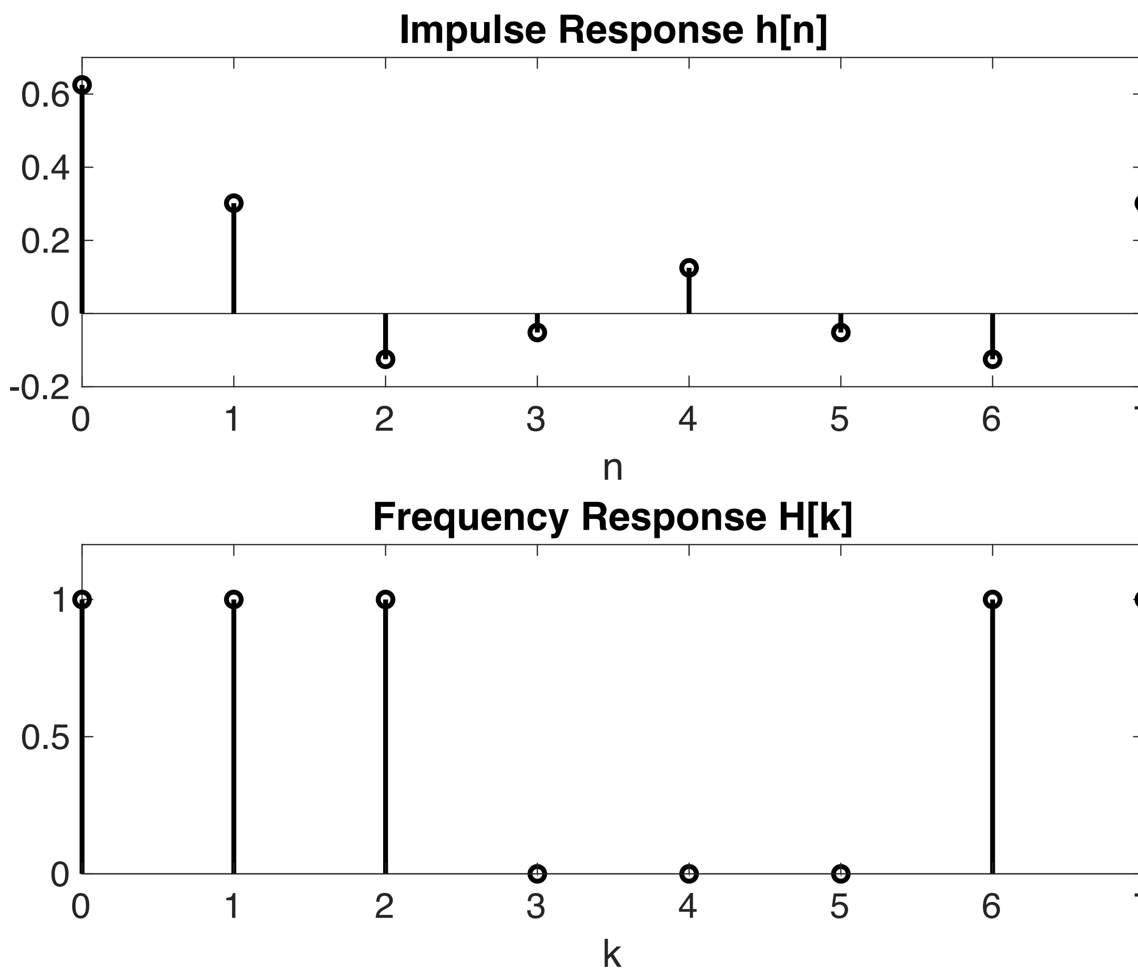
DC offset

Nyquist frequency

Complex conjugates

Better index	0	1	2	3	4 (-4)	-3	-2	-1
Frequency of FFT	0	S/N	2S/N	3S/N	4S/N	-3S/N	-2S/N	-S/N
If S=1000 Hz and N = 8	0	125Hz	250Hz	375Hz	500Hz	-375Hz	-250Hz	-125Hz

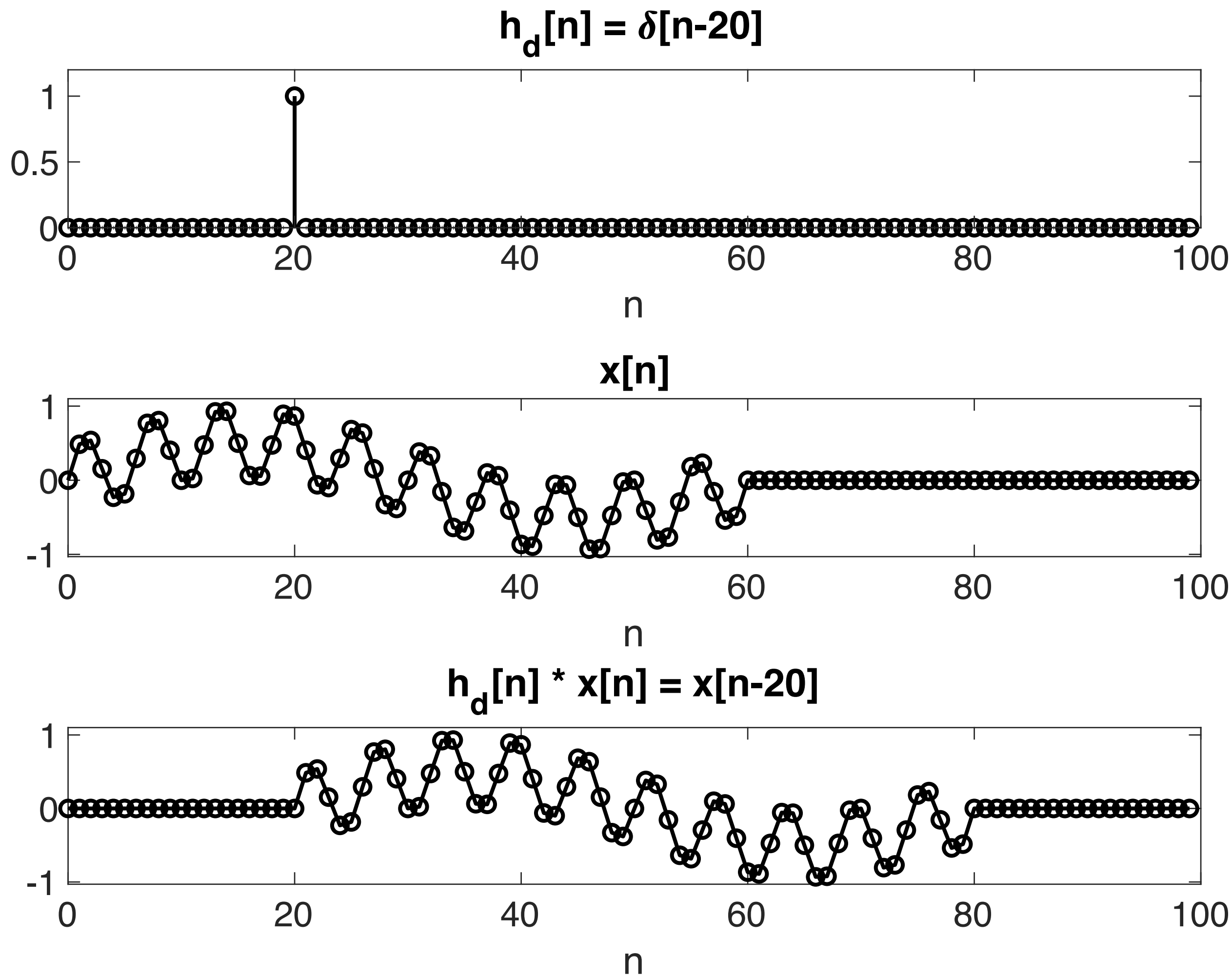
The same 8-point low-pass filter (another view)



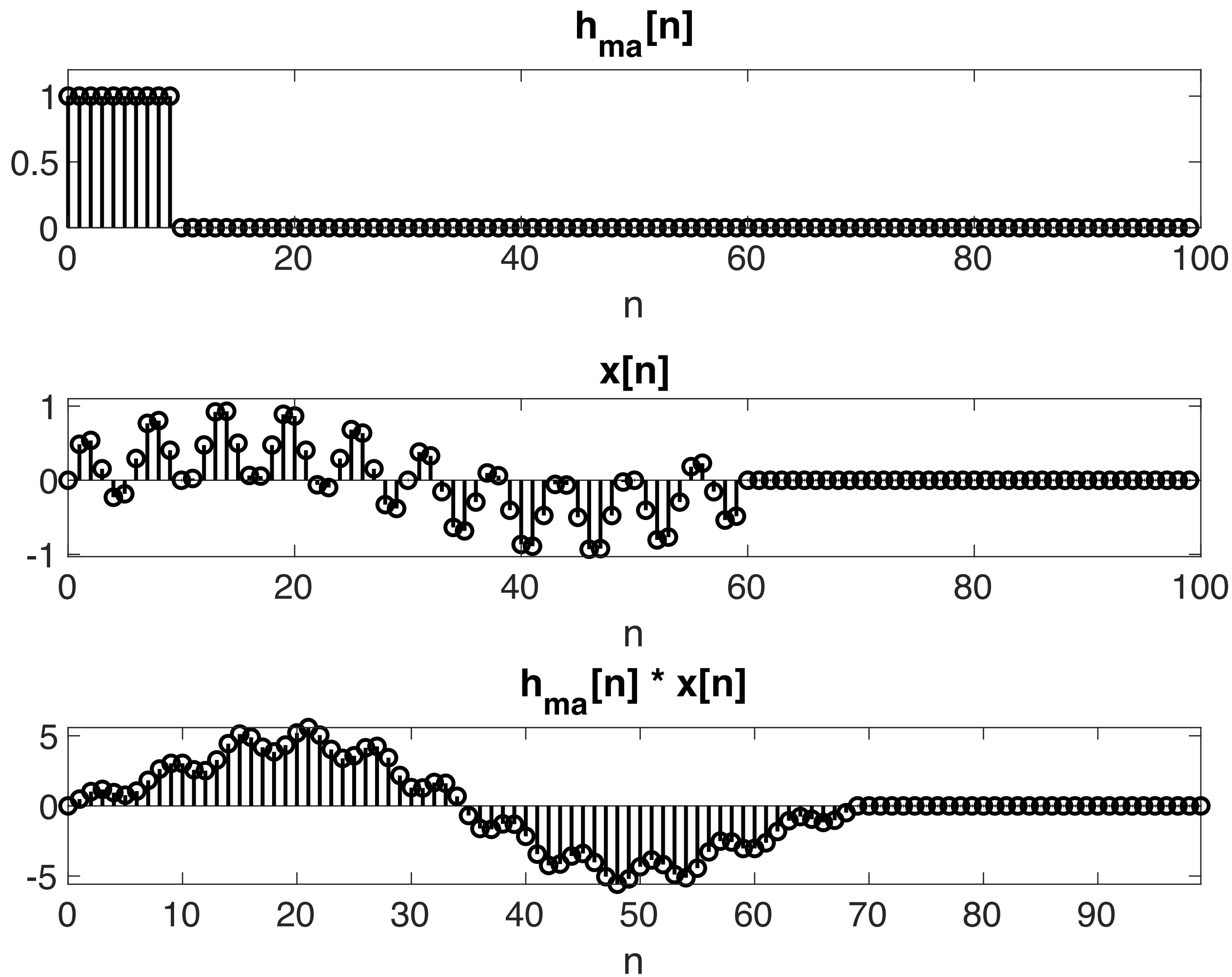
What is reverberation?

- Reverberation is made of echoes
- Echoes are delayed copies of the original sound
- In the physical world these are caused by reflections off of walls and other surfaces
- In the digital world, this is done with impulse response functions and convolution

Example: Delay Operator (aka a single echo)



Several delayed copies = a moving average



Moving average acts as a low-pass filter

- The moving average operator makes copies of the signal and sums them.
- You’re summing multiple, slightly-offset copies.
- This causes high-frequency events to average out in the wash (low-pass filtering).
- What would happen if the copies are offset by the exact period of some frequency in the audio?

A room is a filter

Carnegie Hall

Reverberant Rooms

- …are louder (why?)
- …make speech harder to understand (why?)
- …may emphasize certain frequencies (why)?

They are all strongly related

- A room can be modeled as a **filter**
- **Reverb** is the effect of room/filter on sound signals
- Mathematically, adding reverb to sound signals (filtering) is performed via **convolution**

Making a convolution reverb

- The impulse response function of a room, **h**, captures the echoes (reverberation) that happen when you make a sound in that room.
- You can use **h** to filter an audio recording made elsewhere, **x**.
- If you do that, the echoes from the room will be applied to **x** and it will sound like **x** was recorded in that room.
- This is the basis of **convolution reverb**.

Convolution Reverb in a nutshell

- Record an impulse in the room
- Estimate the impulse response of the room
- Record something in another quiet room (with the microphone up close to the source, so you avoid echoes)
- Apply convolution between your new recording and the impulse response function of the room you want to sound like you recorded in.

Getting h[n]: the impulse response

- Walk into a QUIET room with a recorder
- Turn on the recorder
- Clap your hands ONCE (this is your impulse)
- The recording captures the room's impulse response
- …and other noise (air conditioner, etc.)

Estimating h[n] in a perfect world

- Assume a balloon pop is an impulse. Call it x[n].
- If x[n] is an impulse, then it is a sequence with a 1 at time 0 and a 0 everywhere else.
- Record the balloon pop in a room. Call it y[n].
- If there was no other noise and our recording didn't distort, y[n] is just the convolution of an impulse with h[n].
- This means y[n] = h[n].

Estimating the frequency response *H*

- Given no noise, we can estimate the impulse response h[n] by estimating the frequency response H[k] in the frequency domain

$$X[k]H[k] = Y[k]$$
$$H[k] = Y[k]/ X[k]$$

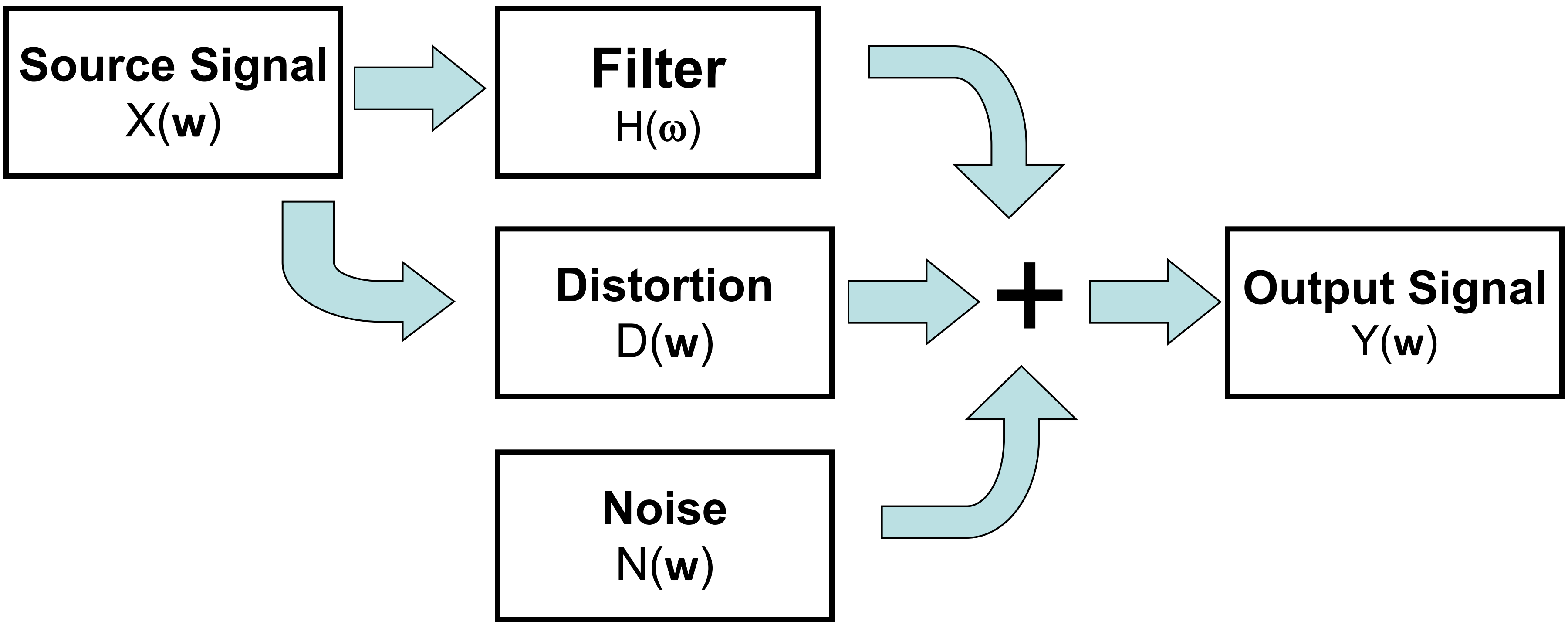
If x[n] is an impulse, then X[k] = 1 for all k. Therefore….

$$H[k] = Y[k]$$

Estimating the frequency response *H*

- But there is noise ...
- What do we do?
- We assume noise is uncorrelated to the signal or the filter
- We assume noise is unbiased (zero mean)
- We try lots of estimates and hope that the noise “washes out” when we average

Noise and Distortion



Distortion and noise make it a lot harder to estimate ***H***

Estimating the frequency response *H*

- Assume we know what X is (because we made it) and what Y is (‘cause we recorded it).
- Hope noise and distortion are not correlated with X.
- Call “noise + distortion” N.
- Then…

$$X[k]H[k]+N[k]=Y[k]$$

Estimating H

- Estimate H a lot of times, in the hopes that the noise will wash out in the mix…

$$\hat{H}[k] \approx \left\langle Y[k]X[k]^* \right\rangle / \left\langle X[k]X[k]^* \right\rangle$$

where

$$\left\langle Y[k]X[k]^* \right\rangle = \frac{1}{N}\sum_{n=1}^{N} Y_n[k]X_n[k]^*$$

This gives the average value over N experiments. Note the * indicates a complex conjugate.

Caveats

- This only works on the frequencies where there was energy in the input signal X.
- If there wasn't energy at a frequency then we're out of luck.
- So, best to use a broadband sound for X.
- Recall: An impulse is broadband.