Linear shift-invariant (LSI) systems

- Linearity (previously discussed):
  “linear combination in, linear combination out”

- Shift-invariance (new property):
  “shifted vector in, shifted vector out”

- Note: These two properties are independent (think of some examples...)
(as before, express input as weighted sum of “impulses”)

(responses to impulses are shifted copies of each other)
LSI systems are characterized by their “impulse response”

\[ y(n) = \sum_k r(n-k)x(k) = \sum_k r(k)x(n-k) \]

- Sliding dot products
- Matrix description
- Boundaries: zero-padded, reflected, circular
- Examples: impulse, delay, average, difference
Feedback LSI system

\[ y(n) = \sum_k f(n-k)x(k) + \sum_k g(n-k)y(k) \]

(In general, we’ll stick to feedforward (FIR) systems)

- Infinite impulse response (IIR)
- Recursive => possibly unstable

2D convolution

- sliding window

[figure c/o Castleman]
Discrete Sinusoids

\[ \cos(\omega n), \quad \omega = \frac{2\pi k}{N} \]

“frequency” (radians/sample)

More generally: \( A \cos(\omega n - \phi) \)

“amplitude”

“phase” (radians)

Shifting Sinusoids

\[ A \cos(\omega n - \phi) = A \cos(\phi) \cos(\omega n) + A \sin(\phi) \sin(\omega n) \]

... using the trigonometric identity:

\[ \cos(a - b) = \cos(a) \cos(b) + \sin(a) \sin(b) \]
A shifted sinusoidal vector can be written as a weighted sum of two fixed sinusoidal vectors!
**Shifting Sinusoids**

\[ A \cos(\omega n - \phi) = A \cos(\phi) \cos(\omega n) + A \sin(\phi) \sin(\omega n) \]

A shifted sinusoidal vector can be written as a weighted sum of two fixed sinusoidal vectors!

**LSI response to sinusoids**

\[ x(n) = \cos(\omega n) \] (input)
LSI response to sinusoids

\[ x(n) = \cos(\omega n) \]

\[ y(n) = \sum_{m} r(m) \cos(\omega(n - m)) \quad \text{(convolution formula)} \]

Inner product of impulse response with cos/sin, respectively

\[ y(n) = \sum_{m} r(m) \cos(\omega m) \cos(\omega n) - \sum_{m} r(m) \sin(\omega m) \sin(\omega n) \quad \text{(trig identity)} \]
LSI response to sinusoids

\[ x(n) = \cos(\omega n) \]

\[ y(n) = \sum_m r(m) \cos(\omega(n - m)) \]

\[ = \sum_m r(m) \cos(\omega m) \cos(\omega n) + \sum_m r(m) \sin(\omega m) \sin(\omega n) \]

\[ = c_r(\omega) \cos(\omega n) + s_r(\omega) \sin(\omega n) \]

(convert rectangular to polar coordinates)

\[ s_r(\omega) \]

\[ A_r(\omega) \]

\[ \phi_r(\omega) \]

\[ c_r(\omega) \]
**LSI response to sinusoids**

\[ x(n) = \cos(\omega n) \]

\[ y(n) = \sum_{m} r(m) \cos(\omega(n - m)) \]

\[ = \sum_{m} r(m) \cos(\omega m) \cos(\omega n) + \sum_{m} r(m) \sin(\omega m) \sin(\omega n) \]

\[ = c_r(\omega) \cos(\omega n) + s_r(\omega) \sin(\omega n) \]

\[ = A_r(\omega) \cos(\phi_r(\omega)) \cos(\omega n) + A_r(\omega) \sin(\phi_r(\omega)) \sin(\omega n) \]

\[ = A_r(\omega) \cos(\omega n - \phi_r(\omega)) \quad \text{(trig identity, in the opposite direction)} \]

“Sinusoid in, sinusoid out” (with modified amplitude/phase)

---

**LSI response to sinusoids**

More generally, if input has amplitude \( A_x \) and phase \( \phi_x \),

\[ x(n) = A_x \cos(\omega n - \phi_x) \]

\[ y(n) = A_r(\omega) A_x \cos(\omega n - \phi_x - \phi_r(\omega)) \]

amplitudes multiply \hspace{1cm} \text{phases add}

“Sinusoid in, sinusoid out” (with modified amplitude/phase)
Discrete Fourier transform (DFT)

- Construct an orthogonal matrix of sin/cos pairs, at frequency multiples of $\frac{2\pi}{N}$ radians/sample, (i.e., $2\pi k/N$, for $k = 0, 1, 2, \ldots N/2$)
- For $k = 0$ and $k = N/2$, only need the cosine part (thus, $N/2+1$ cosines, and $N/2-1$ sines)
- When we apply this matrix to an input vector, think of output as paired coordinates
- Common to plot these pairs as amplitude/phase

[all details on board...]

The Fourier family

<table>
<thead>
<tr>
<th>frequency domain</th>
<th>continuous</th>
<th>discrete</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuous</td>
<td>Fourier transform</td>
<td>discrete-time Fourier transform</td>
</tr>
<tr>
<td>discrete</td>
<td>Fourier series</td>
<td><strong>discrete Fourier transform</strong></td>
</tr>
</tbody>
</table>

The “fast Fourier transform” (FFT) is a computationally efficient implementation of the DFT (cost with vector length as $N \log(N)$, instead of $N^2$).
LSI response to sinusoids

\[ x(n) = \cos(\omega n) \]

\[ y(n) = \sum_m r(m) \cos (\omega (n - m)) \]

\[ = \sum_m r(m) \cos(\omega m) \cos(\omega n) + \sum_m r(m) \sin(\omega m) \sin(\omega n) \]

\[ = c_r(\omega) \cos(\omega n) + s_r(\omega) \sin(\omega n) \]

\[ = A_r(\omega) \cos(\phi_r(\omega)) \cos(\omega n) + A_r(\omega) \sin(\phi_r(\omega)) \sin(\omega n) \]

\[ = A_r(\omega) \cos(\omega n - \phi_r(\omega)) \]

NOTE: These dot products are just the Fourier transform of the impulse response \( r(m) \)!
Fourier & LSI

\[ \vec{x} \rightarrow L \]

\[ c_x(0) \]
\[ c_x(1) \]
\[ s_x(1) \]
\[ c_x(2) \]
\[ s_x(2) \]

note: only 3 (of many) frequency components shown

Fourier & LSI

\[ \vec{x} \rightarrow L \]

\[ A_x(0) \]
\[ \phi_x(1) \]
\[ A_x(1) \]
\[ \phi_x(2) \]
\[ A_x(2) \]

note: only 3 (of many) frequency components shown
LSI systems are characterized by their *frequency response*, specified by the Fourier Transform of their impulse response.

**Complex exponentials:**

“bundling” sine and cosine

\[ e^{i\theta} = \cos(\theta) + i \sin(\theta) \]

\[ e^{i\omega n} \rightarrow L \rightarrow A_r(\omega) e^{i(\omega n-\phi_r(\omega))} = A_r(\omega) e^{-i\phi_r(\omega)} e^{i\omega n} = \bar{f}(\omega) e^{i\omega n} \]
Complex exponentials: “bundling” sine and cosine

\[ e^{i\theta} = \cos(\theta) + i \sin(\theta) \]

\[ e^{i\omega n} \rightarrow L \rightarrow A_r(\omega) e^{i(\omega n - \phi_r(\omega))} = A_r(\omega) e^{-i\phi_r(\omega)} e^{i\omega n} = \tilde{r}(\omega) e^{i\omega n} \]

Note: implies that complex exponentials are eigenvectors!

F.T. of impulse response!
The “convolution theorem”

\[ \tilde{x} \quad \text{convolve with} \quad \tilde{r} \quad \longrightarrow \quad \tilde{y} \]

The “convolution theorem”

\[ \tilde{x} \quad \text{convolve with} \quad \tilde{r} \quad \longrightarrow \quad \tilde{y} \]

Fourier Transform

\[ \tilde{x} \quad \text{pointwise multiply by} \quad \tilde{r} \quad \longrightarrow \quad \tilde{y} \]

Fourier Transform
The “convolution theorem”

\[ \mathbf{y} = \mathbf{Lx} = \mathbf{F} \mathbf{\tilde{R}} \mathbf{F}^T \mathbf{x} \quad \Rightarrow \quad \mathbf{F}^T \mathbf{y} = \mathbf{\tilde{R}} \mathbf{F}^T \mathbf{x} \]

(diagonal matrix)

Recap

- Linear system
  => defined by superposition
  => characterized by a matrix

- Linear Shift-invariant (LSI) system
  => defined by superposition and shift-invariance
  => characterized by a single impulse response
  => alternatively, characterized by frequency response (the Fourier Transform of the impulse response!), which specifies an amplitude multiplier and a phase shift.
What do we do with Fourier Transforms?

Useful for representing/analyzing periodic signals

Eigenvectors of LSI systems => useful for analysis/design of these systems. In particular, how do you identify the nullspace?

Discrete Fourier transform (with complex numbers)

\[
\tilde{r}_k = \sum_{n=0}^{N-1} r_n e^{-i\omega_k n} \quad \text{where} \quad \omega_k = \frac{2\pi k}{N}
\]

\[
r_n = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{r}_k e^{i\omega_k n} \quad \text{(inverse)}
\]
Visualizing the (discrete) Fourier transform

- Two conventional choices for frequency axis:
  - Plot frequencies from $k=0$ to $k=N/2$
  - Plot frequencies from $k=-N/2$ to $N/2-1$

- Typically, plot Amplitude (and possibly Phase, on a separate graph), instead of cosine/sine (real/imaginary) parts

\[ e^{i\omega n} = \cos(\omega n) + i \sin(\omega n) \]
\[ \cos(\omega n) = \frac{1}{2}(e^{i\omega n} + e^{-i\omega n}) \]
\[ \Rightarrow \]
\[ \sin(\omega n) = \frac{-i}{2}(e^{i\omega n} - e^{-i\omega n}) \]

Example for $k=2$, $N=32$ (note indexing and amplitudes):
More examples

• constant
• sinusoid (see next slide)
• impulse
• Gaussian - “lowpass”
• DoG (difference of 2 Gaussians) - “bandpass”
• Gabor (Gaussian windowed sinusoid) - “bandpass”

Retinal ganglion cells (1D)

Enroth-Cugell and Robson (1984)
Sampling causes “aliasing”

Sampling process is linear, but many-to-one (non-invertible)

“Aliasing” - one frequency masquerades as another

Given the samples, it is common/natural to assume that they arose from the lowest compatible frequency...

Effect of sampling on the Fourier Transform:
Sum of shifted copies

\[ X(\omega) \]

\[ X_s(\omega) \]
Real-world aliasing

downsampling by 2

Pre-filtering to avoid spectral overlap ("aliasing")

\[ X(\omega) \xrightarrow{L(\omega)} L(\omega) \xrightarrow{\Delta} X_s(\omega) \]

\[ X(\omega) \]

\[ L(\omega) \quad \text{lowpass filter, cutoff at } \pi/\Delta \]

\[ X_s(\omega) \]
Real-world aliasing

downsampling by 2, with pre-filtering