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...)

Section 3:
Linear Shift-invariant Systems
As before, express input as a sum of “impulses”, weighted by elements of $x$.

• Linearity $\Rightarrow$ response to $x$ is sum of responses to impulses, weighted by elements of $x$

• Shift-invariance $\Rightarrow$ responses to impulses are shifted copies of each other
LSI system

LSI systems are characterized by their "impulse response"

Convolution

$$y(n) = \sum_{k} r(n - k)x(k)$$

- 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 \(\Rightarrow\) possibly unstable

2D convolution

- sliding window
Discrete Sinusoids

\[ \cos(\omega n), \quad \omega = 2\pi k/N \]

More generally:

\[ A \cos(\omega n - \phi) \]

Shifting Sinusoids

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

... via a well-known trigonometric identity:

\[ \cos(a - b) = \cos(a) \cos(b) + \sin(a) \sin(b) \]

We’ll also need conversions between polar and rectangular coordinates:

\[ x = A \cos(\phi), \quad y = A \sin(\phi) \]

\[ A = \sqrt{x^2 + y^2}, \quad \phi = \tan^{-1}(y/x) \]
Shifting Sinusoids

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

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

LSI response to sinusoids

\[ x(n) = \cos(\omega n) \quad \text{(input)} \]
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
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)
LSI response to sinusoids

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

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

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

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

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

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

“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 phases add

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

- Construct an orthogonal matrix of sin/cos pairs, covering different numbers of cycles
- Frequency multiples of $2\pi/N$ radians/sample, (specifically, $2\pi k/N$, 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

\[ F = \begin{bmatrix}
F_{k=0} & F_{k=1} & F_{k=2} & F_{k=3} & \cdots
\end{bmatrix} \]
The Fourier family

**signal domain**

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

Note: the “fast Fourier transform” (FFT) is a computationally efficient implementation of the DFT. Computational cost is $N \log(N)$ operations, compared to the $N^2$ operations that would be needed for matrix multiplication.

**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
LSI systems are characterized by their frequency response, specified by the Fourier Transform of their impulse response,
Complex exponentials: “bundling” sine and cosine

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

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

real part:

imaginary part:

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

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

\[ = \tilde{r}(\omega) e^{i\omega n} \]

F.T. of impulse response!

Note: implies that complex exponentials are eigenvectors!
The “convolution theorem”

\[ \tilde{x} \rightarrow \tilde{y} \]
convolve with \( \tilde{r} \)

\[ \tilde{x} \rightarrow \tilde{y} \]
pointwise multiply by \( \tilde{r} \)

\[ \tilde{x} \rightarrow \tilde{y} \]
Fourier Transform

\[ \tilde{x} \rightarrow \tilde{y} \]
inverse Fourier Transform
The “convolution theorem”

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

(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 vector (the impulse response)
  - alternatively, characterized by frequency response (the Fourier Transform of the impulse response), which specifies an amplitude multiplier and a phase shift for each frequency.
What do we do with Fourier Transforms?

- Represent/analyze periodic \textit{signals}

- Analyze/design LSI \textit{systems}. In particular, how do you identify the nullspace?

\textit{Discrete} Fourier transform
(with complex numbers)

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

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

[on board: why minus sign? why 1/N?]
Visualizing the (Discrete) Fourier transform

- Two conventional choices for frequency axis:
  - Plot frequencies from $k=0$ to $k=N/2$
    (in matlab: 1 to $N/2-1$)
  - Plot frequencies from $k=-N/2$ to $N/2-1$
    (in matlab: use fftshift)

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

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

$$e^{i\omega n} = \cos(\omega n) + i \sin(\omega n)$$

$$\cos(\omega n) = \frac{1}{2}(e^{i\omega n} + e^{-i\omega n})$$

$$\Rightarrow$$

$$\sin(\omega n) = \frac{-i}{2}(e^{i\omega n} - e^{-i\omega n})$$

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”

[on board]

Retinal ganglion cells (1D)
Sampling causes “aliasing”

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

“Aliasing” - one frequency masquerades as another \[on\ \text{board}\]

Given the samples, it is common/natural to assume that they arose from the \textit{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

“Moiré pattern”

Pre-filtering to avoid spectral overlap (“aliasing”)

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

\[ |X(\omega)| \]

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

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

downsample by 2, with pre-filtering