Systems

Systems are Transformations

DEFINITION

A discrete-time **system** $\mathcal{H}$ is a transformation (a rule or formula) that maps a discrete-time input signal x into a discrete-time output signal y

$$y = \mathcal{H}\{x\}$$

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

- Systems manipulate the information in signals
- Examples:
  - A speech recognition system converts acoustic waves of speech into text
  - A radar system transforms the received radar pulse to estimate the position and velocity of targets
  - A functional magnetic resonance imaging (fMRI) system transforms measurements of electron spin into voxel-by-voxel estimates of brain activity
  - A 30 day moving average smooths out the day-to-day variability in a stock price

Signal Length and Systems

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

- Recall that there are two kinds of signals: <u>infinite-length</u> and <u>finite-length</u>

- Accordingly, we will consider two kinds of systems:

  1. Systems that transform an infinite-length-signal x into an infinite-length signal y
  2. Systems that transform a length-N signal x into a length-N signal y

     (Such systems can also be used to process periodic signals with period N)

- For generality, we will assume that the input and output signals are complex valued

System Examples (1)

- Identity

$$y[n] = x[n] \quad \forall n$$

- Scaling

$$y[n] = 2\,x[n] \quad \forall n$$

- Offset

$$y[n] = x[n] + 2 \quad \forall n$$

- Square signal

$$y[n] = (x[n])^2 \quad \forall n$$

- Shift

$$y[n] = x[n+2] \quad \forall n$$

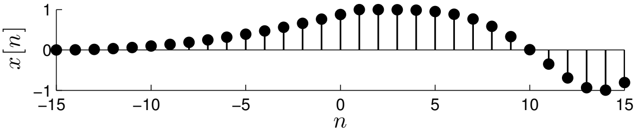
- Decimate

$$y[n] = x[2n] \quad \forall n$$

- Square time

$$y[n] = x[n^2] \quad \forall n$$

System Examples (2)



- Shift system ($m \in \mathbb{Z}$ fixed)

$$y[n] = x[n-m] \quad \forall n$$

- Moving average (combines shift, sum, scale)

$$y[n] = \frac{1}{2}(x[n] + x[n-1]) \quad \forall n$$

- Recursive average

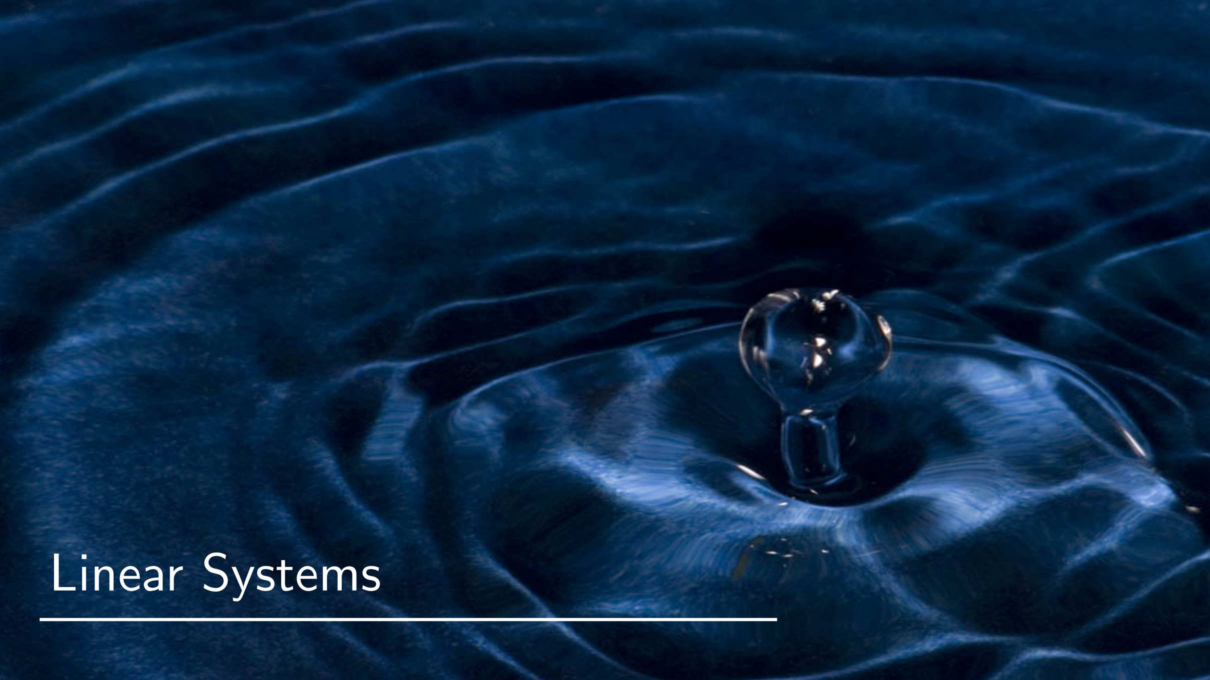
$$y[n] = x[n] + \alpha\, y[n-1] \quad \forall n$$

Summary

- Systems transform one signal into another to manipulate information
- We will consider two kinds of systems:
  1. Systems that transform an infinite-length-signal x into an infinite-length signal y
  2. Systems that transform a length-N signal x into a length-N signal y

     (Such systems can also be used to process periodic signals with period N)

Linear Systems



Linear Systems

DEFINITION

A system $\mathcal{H}$ is (zero-state) **linear** if it satisfies the following two properties:

1. Scaling

$$\mathcal{H}\{\alpha\, x\} = \alpha\, \mathcal{H}\{x\} \quad \forall\, \alpha \in \mathbb{C}$$

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y \qquad\qquad \alpha\, x \longrightarrow \boxed{\mathcal{H}} \longrightarrow \alpha\, y$$

2. Additivity

$$\text{If} \quad y_1 = \mathcal{H}\{x_1\} \quad \text{and} \quad y_2 = \mathcal{H}\{x_2\} \quad \text{then}$$

$$\mathcal{H}\{x_1 + x_2\} = y_1 + y_2$$

$$x_1 \longrightarrow \boxed{\mathcal{H}} \longrightarrow y_1 \qquad\qquad x_2 \longrightarrow \boxed{\mathcal{H}} \longrightarrow y_2$$

$$x_1 + x_2 \longrightarrow \boxed{\mathcal{H}} \longrightarrow y_1 + y_2$$

Linearity Notes

- A system that is not linear is called **nonlinear**
- To prove that a system is linear, you must prove rigorously that it has **both** the scaling and additivity properties for **arbitrary** input signals
- To prove that a system is nonlinear, it is sufficient to exhibit a **counterexample**

Example: Moving Average is Linear (Scaling)

$$x[n] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n] = \tfrac{1}{2}(x[n] + x[n-1])$$

- **Scaling:** (Strategy to prove – Scale input x by $\alpha \in \mathbb{C}$, compute output y via the formula at top, and verify that it is scaled as well)
  - Let
  $$x'[n] = \alpha x[n], \quad \alpha \in \mathbb{C}$$
  - Let y' denote the output when x' is input (that is, $y' = \mathcal{H}\{x'\}$)
  - Then
  $$y'[n] = \frac{1}{2}(x'[n] + x'[n-1]) = \frac{1}{2}(\alpha x[n] + \alpha x[n-1]) = \alpha\left(\frac{1}{2}(x[n] + x[n-1])\right) = \alpha y[n] \ \checkmark$$

Example: Moving Average is Linear (Additivity)

$$x[n] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n] = \tfrac{1}{2}(x[n] + x[n-1])$$

- **Additivity:** (Strategy to prove – Input two signals into the system and verify that the output equals the sum of the respective outputs)
  - Let
$$x'[n] \;=\; x_1[n] + x_2[n]$$
  - Let $y'/y_1/y_2$ denote the output when $x'/x_1/x_2$ is input
  - Then
$$\begin{aligned} y'[n] &= \frac{1}{2}(x'[n] + x'[n-1]) \;=\; \frac{1}{2}(\{x_1[n] + x_2[n]\} + \{x_1[n-1] + x_2[n-1]\}) \\ &= \frac{1}{2}(x_1[n] + x_1[n-1]) + \frac{1}{2}(x_2[n] + x_2[n-1]) \;=\; y_1[n] + y_2[n] \quad \checkmark \end{aligned}$$

Example: Squaring is Nonlinear

$$x[n] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n] = (x[n])^2$$

- **Additivity:** Input two signals into the system and see what happens
  - Let
  $$y_1[n] = (x_1[n])^2\,, \qquad y_2[n] = (x_2[n])^2$$
  - Set
  $$x'[n] \;=\; x_1[n] + x_2[n]$$
  - Then
  $$y'[n] \;=\; \left(x'[n]\right)^2 \;=\; (x_1[n] + x_2[n])^2 \;=\; (x_1[n])^2 + 2x_1[n]x_2[n] + (x_2[n])^2 \;\neq\; y_1[n] + y_2[n]$$
  - Nonlinear!

Linear or Nonlinear? You Be the Judge! (1)

- Identity

$$y[n] = x[n] \quad \forall n$$

- Scaling

$$y[n] = 2\,x[n] \quad \forall n$$

- Offset

$$y[n] = x[n] + 2 \quad \forall n$$

- Square signal

$$y[n] = (x[n])^2 \quad \forall n$$

- Shift

$$y[n] = x[n+2] \quad \forall n$$

- Decimate

$$y[n] = x[2n] \quad \forall n$$

- Square time

$$y[n] = x[n^2] \quad \forall n$$

Linear or Nonlinear? You Be the Judge! (2)

- Shift system ($m \in \mathbb{Z}$ fixed)

$$y[n] = x[n-m] \quad \forall n$$

- Moving average (combines shift, sum, scale)

$$y[n] = \frac{1}{2}(x[n] + x[n-1]) \quad \forall n$$

- Recursive average

$$y[n] = x[n] + \alpha\, y[n-1] \quad \forall n$$

Matrix Multiplication and Linear Systems

- Matrix multiplication (aka Linear Combination) is a fundamental signal processing system
- **Fact 1:** Matrix multiplications are linear systems (easy to show at home, but do it!)

$$y = \mathbf{H}\, x$$

$$y[n] = \sum_m [\mathbf{H}]_{n,m}\, x[m]$$

  (Note: This formula applies for both infinite-length and finite-length signals)

- **Fact 2:** *All* linear systems can be expressed as matrix multiplications
- As a result, we will use the matrix viewpoint of linear systems extensively in the sequel
- Try at home: Express all of the linear systems in the examples above in matrix form

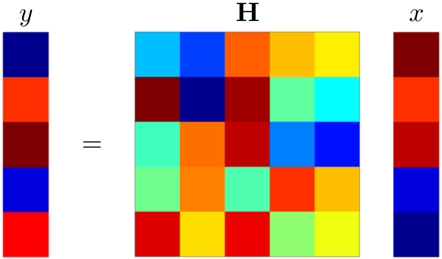
Matrix Multiplication and Linear Systems in Pictures

- Linear system

$$y \;=\; \mathbf{H}\,x$$

$$y[n] \;=\; \sum_m [\mathbf{H}]_{n,m}\,x[m] \;=\; \sum_m h_{n,m}\,x[m]$$

where $h_{n,m} \;=\; [\mathbf{H}]_{n,m}$ represents the row-n, column-m entry of the matrix $\mathbf{H}$



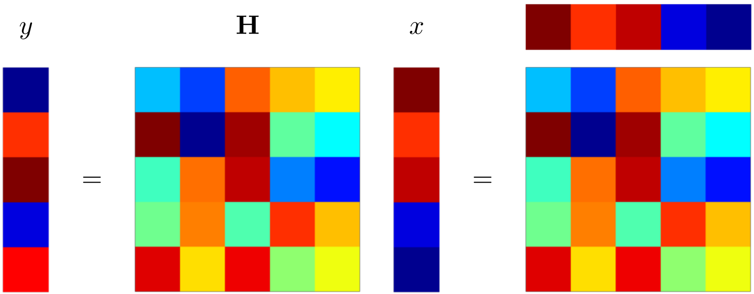
System Output as a Linear Combination of Columns

- Linear system

$$y \;=\; \mathbf{H}\,x$$

$$y[n] \;=\; \sum_m [\mathbf{H}]_{n,m}\,x[m] \;=\; \sum_m h_{n,m}\,x[m]$$

where $h_{n,m} \;=\; [\mathbf{H}]_{n,m}$ represents the row-n, column-m entry of the matrix $\mathbf{H}$



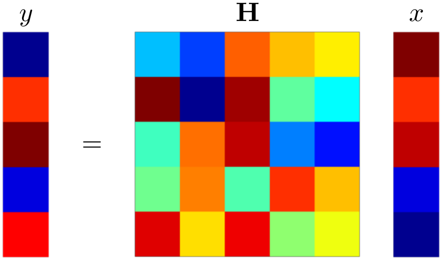
System Output as a Sequence of Inner Products

- Linear system

$$y = \mathbf{H}\, x$$

$$y[n] = \sum_{m} [\mathbf{H}]_{n,m}\, x[m] = \sum_{m} h_{n,m}\, x[m]$$

where $h_{n,m} = [\mathbf{H}]_{n,m}$ represents the row-n, column-m entry of the matrix $\mathbf{H}$



Summary

- Linear systems satisfy (1) scaling and (2) additivity
- To show a system is <u>linear</u>, you have to prove it rigorously assuming arbitrary inputs (work!)
- To show a system is <u>nonlinear</u>, you can just exhibit a counterexample (often easy!)
- Linear systems $\equiv$ matrix multiplication
  - Justifies our emphasis on linear vector spaces and matrices
  - The output signal y equals the linear combination of the columns of $\mathbf{H}$ weighted by the entries in x
  - Alternatively, the output value $y[n]$ equals the inner product between row n of $\mathbf{H}$ with x

Time-Invariant Systems

Time-Invariant Systems (Infinite-Length Signals)

DEFINITION

A system $\mathcal{H}$ processing infinite-length signals is **time-invariant** (shift-invariant) if a time shift of the input signal creates a corresponding time shift in the output signal

$$x[n] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n]$$

$$x[n-q] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n-q]$$

- Intuition: A time-invariant system behaves the same no matter when the input is applied
- A system that is not time-invariant is called **time-varying**

Example: Moving Average is Time-Invariant

$$x[n] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n] = \tfrac{1}{2}(x[n] + x[n-1])$$

- Let

$$x'[n] = x[n-q], \quad q \in \mathbb{Z}$$

- Let y' denote the output when x' is input (that is, $y' = \mathcal{H}\{x'\}$)

- Then

$$y'[n] \;=\; \frac{1}{2}(x'[n] + x'[n-1]) \;=\; \frac{1}{2}(x[n-q] + x[n-q-1]) \;=\; y[n-q] \quad \checkmark$$

Example: Decimation is Time-Varying

$$x[n] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n] = x[2n]$$

- This system is time-varying; demonstrate with a counter-example
- Let
$$x'[n] = x[n-1]$$
- Let y' denote the output when x' is input (that is, $y' = \mathcal{H}\{x'\}$)
- Then
$$y'[n] \ = \ x'[2n] \ = \ x[2n-1] \ \neq \ x[2(n-1)] \ = \ y[n-1]$$

Time-Invariant or Time-Varying? You Be the Judge! (1)

- Identity

$$y[n] = x[n] \quad \forall n$$

- Scaling

$$y[n] = 2\,x[n] \quad \forall n$$

- Offset

$$y[n] = x[n] + 2 \quad \forall n$$

- Square signal

$$y[n] = (x[n])^2 \quad \forall n$$

- Shift

$$y[n] = x[n+2] \quad \forall n$$

- Decimate

$$y[n] = x[2n] \quad \forall n$$

- Square time

$$y[n] = x[n^2] \quad \forall n$$

Time-Invariant or Time-Varying? You Be the Judge! (2)

- Shift system ($m \in \mathbb{Z}$ fixed)

$$y[n] = x[n-m] \quad \forall n$$

- Moving average (combines shift, sum, scale)

$$y[n] = \frac{1}{2}(x[n] + x[n-1]) \quad \forall n$$

- Recursive average

$$y[n] = x[n] + \alpha\, y[n-1] \quad \forall n$$

Time-Invariant Systems (Finite-Length Signals)

DEFINITION

A system $\mathcal{H}$ processing length-N signals is **time-invariant** (shift-invariant) if a circular time shift of the input signal creates a corresponding circular time shift in the output signal

$$x[n] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n]$$

$$x[(n-q)_N] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[(n-q)_N]$$

- Intuition: A time-invariant system behaves the same no matter when the input is applied
- A system that is not time-invariant is called **time-varying**

Summary

- Time-invariant systems behave the same no matter when the input is applied
- Infinite-length signals: Invariance with respect to any integer time shift
- Finite-length signals: Invariance with respect to a circular time shift
- To show a system is <u>time-invariant</u>, you have to prove it rigorously assuming arbitrary inputs (work!)
- To show a system is <u>time-varying</u>, you can just exhibit a counterexample (often easy!)

Linear Time-Invariant Systems

Linear Time Invariant (LTI) Systems

DEFINITION

A system $\mathcal{H}$ is **linear time-invariant** (LTI) if it is both linear and time-invariant

- LTI systems are the foundation of signal processing and the main subject of this course

LTI or Not? You Be the Judge! (1)

- Identity

$$y[n] = x[n] \quad \forall n$$

- Scaling

$$y[n] = 2\,x[n] \quad \forall n$$

- Offset

$$y[n] = x[n] + 2 \quad \forall n$$

- Square signal

$$y[n] = (x[n])^2 \quad \forall n$$

- Shift

$$y[n] = x[n+2] \quad \forall n$$

- Decimate

$$y[n] = x[2n] \quad \forall n$$

- Square time

$$y[n] = x[n^2] \quad \forall n$$

LTI or Not? You Be the Judge! (2)

- Shift system ($m \in \mathbb{Z}$ fixed)

$$y[n] = x[n-m] \quad \forall n$$

- Moving average (combines shift, sum, scale)

$$y[n] = \frac{1}{2}(x[n] + x[n-1]) \quad \forall n$$

- Recursive average

$$y[n] = x[n] + \alpha\, y[n-1] \quad \forall n$$

Matrix Multiplication and LTI Systems (Infinite-Length Signals)

- Recall that all linear systems can be expressed as matrix multiplications

$$y = \mathbf{H}\,x$$

$$y[n] = \sum_m [\mathbf{H}]_{n,m}\, x[m]$$

  Here $\mathbf{H}$ is a matrix with infinitely many rows and columns

- Let $h_{n,m} = [\mathbf{H}]_{n,m}$ represent the row-n, column-m entry of the matrix $\mathbf{H}$

$$y[n] = \sum_m h_{n,m}\, x[m]$$

- When the <u>linear system</u> is also <u>shift invariant</u>, $\mathbf{H}$ has a special structure

Matrix Structure of LTI Systems (Infinite-Length Signals)

- Linear system for infinite-length signals can be expressed as

$$y[n] \;=\; \mathcal{H}\{x[n]\} \;=\; \sum_{m=-\infty}^{\infty} h_{n,m}\, x[m], \qquad -\infty < n < \infty$$

- Enforcing time invariance implies that for all $q \in \mathbb{Z}$

$$\mathcal{H}\{x[n-q]\} \;=\; \sum_{m=-\infty}^{\infty} h_{n,m}\, x[m-q] \;=\; y[n-q]$$

- Change of variables: $n' = n - q$ and $m' = m - q$

$$\mathcal{H}\{x[n']\} \;=\; \sum_{m'=-\infty}^{\infty} h_{n'+q,m'+q}\, x[m'] \;=\; y[n']$$

- Comparing first and third equations, we see that for an LTI system

$$h_{n,m} \;=\; h_{n+q,m+q} \quad \forall\, q \in \mathbb{Z}$$

LTI Systems are Toeplitz Matrices (Infinite-Length Signals) (1)

- For an LTI system with infinite-length signals

$$h_{n,m} \;=\; h_{n+q,m+q} \quad \forall\, q \in \mathbb{Z}$$

$$\mathbf{H} \;=\; \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h_{-1,-1} & h_{-1,0} & h_{-1,1} & \cdots \\ \cdots & h_{0,-1} & h_{0,0} & h_{0,1} & \cdots \\ \cdots & h_{1,-1} & h_{1,0} & h_{1,1} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix} \;=\; \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h_{0,0} & h_{-1,0} & h_{-2,0} & \cdots \\ \cdots & h_{1,0} & h_{0,0} & h_{-1,0} & \cdots \\ \cdots & h_{2,0} & h_{1,0} & h_{0,0} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$$

- Entries on the matrix diagonals are the same – **Toeplitz matrix**

LTI Systems are Toeplitz Matrices (Infinite-Length Signals) (2)

- All of the entries in a Toeplitz matrix can be expressed in terms of the entries of the
  - 0-**th column**: $h[n] = h_{n,0}$
  - **Time-reversed** 0-**th row**: $h[m] = h_{0,-m}$

$$\mathbf{H} = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h_{0,0} & h_{-1,0} & h_{-1,1} & \cdots \\ \cdots & h_{1,0} & h_{0,0} & h_{-1,0} & \cdots \\ \cdots & h_{2,0} & h_{1,0} & h_{0,0} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix} = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h[0] & h[-1] & h[-2] & \cdots \\ \cdots & h[1] & h[0] & h[-1] & \cdots \\ \cdots & h[2] & h[1] & h[0] & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$$

- Row-n, column-m entry of the matrix $[\mathbf{H}]_{n,m} = h_{n,m} = h[n-m]$

LTI Systems are Toeplitz Matrices (Infinite-Length Signals) (3)

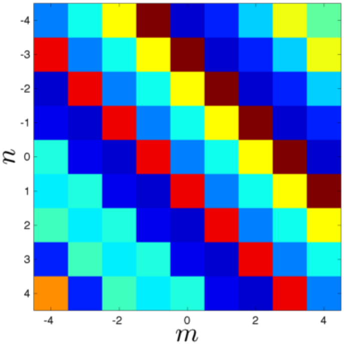
- All of the entries in a Toeplitz matrix can be expressed in terms of the entries of the
  - 0-**th column**: $h[n] = h_{n,0}$ (this is an infinite-length signal/column vector; call it h)
  - **Time-reversed 0-th row**: $h[m] = h_{0,-m}$

- Example: Snippet of a Toeplitz matrix

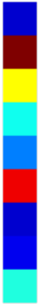
$$
\begin{aligned}
[\mathbf{H}]_{n,m} &= h_{n,m} \\
&= h[n-m]
\end{aligned}
$$

- Note the diagonals!

$\mathbf{H} =$



$h =$



Matrix Structure of LTI Systems (Finite-Length Signals)

- Linear system for signals of length N can be expressed as

$$y[n] \;=\; \mathcal{H}\{x[n]\} \;=\; \sum_{m=0}^{N-1} h_{n,m}\, x[m], \quad 0 \le n \le N-1$$

- Enforcing time invariance implies that for all $q \in \mathbb{Z}$

$$\mathcal{H}\{x[(n-q)_N]\} \;=\; \sum_{m=0}^{N-1} h_{n,m}\, x[(m-q)_N] \;=\; y[(n-q)_N]$$

- Change of variables: $n' = n - q$ and $m' = m - q$

$$\mathcal{H}\{x[(n')_N]\} \;=\; \sum_{m'=-q}^{M-1-q} h_{(n'+q)_N,(m'+q)_N}\, x[(m')_N] \;=\; y[(n')_N]$$

- Comparing first and third equations, we see that for an LTI system

$$h_{n,m} \;=\; h_{(n+q)_N,(m+q)_N} \quad \forall\, q \in \mathbb{Z}$$

LTI Systems are Circulent Matrices (Finite-Length Signals) (1)

- For an LTI system with length-N signals

$$h_{n,m} \;=\; h_{(n+q)_N,(m+q)_N} \quad \forall\, q \in \mathbb{Z}$$

$$\begin{bmatrix} h_{0,0} & h_{0,1} & h_{0,2} & \cdots & h_{0,N-1} \\ h_{1,0} & h_{1,1} & h_{1,2} & \cdots & h_{1,N-1} \\ h_{2,0} & h_{2,1} & h_{2,2} & \cdots & h_{2,N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ h_{N-1,0} & h_{N-1,1} & h_{N-1,2} & \cdots & h_{N-1,N-1} \end{bmatrix} = \begin{bmatrix} h_{0,0} & h_{N-1,0} & h_{N-2,0} & \cdots & h_{1,0} \\ h_{1,0} & h_{0,0} & h_{N-1,0} & \cdots & h_{2,0} \\ h_{2,0} & h_{1,0} & h_{0,0} & \cdots & h_{3,0} \\ \vdots & \vdots & \vdots & & \vdots \\ h_{N-1,0} & h_{N-2,0} & h_{N-3,0} & \cdots & h_{0,0} \end{bmatrix}$$

- Entries on the matrix diagonals are the same $+$ circular wraparound – **circulent matrix**

LTI Systems are Circulent Matrices (Finite-Length Signals) (2)

- All of the entries in a circulent matrix can be expressed in terms of the entries of the
  - 0-**th column**: $h[n] = h_{n,0}$
  - **Circularly time-reversed** 0**-th row**: $h[m] = h_{0,(-m)_N}$

$$\begin{bmatrix} h_{0,0} & h_{N-1,0} & h_{N-2,0} & \cdots & h_{1,0} \\ h_{1,0} & h_{0,0} & h_{N-1,0} & \cdots & h_{2,0} \\ h_{2,0} & h_{1,0} & h_{0,0} & \cdots & h_{3,0} \\ \vdots & \vdots & \vdots & & \vdots \\ h_{N-1,0} & h_{N-2,0} & h_{N-3,0} & \cdots & h_{0,0} \end{bmatrix} = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ h[2] & h[1] & h[0] & \cdots & h[3] \\ \vdots & \vdots & \vdots & & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \cdots & h[0] \end{bmatrix}$$

- Row-n, column-m entry of the matrix $[\mathbf{H}]_{n,m} = h_{n,m} = h[(n-m)_N]$

LTI Systems are Circulent Matrices (Finite-Length Signals) (3)

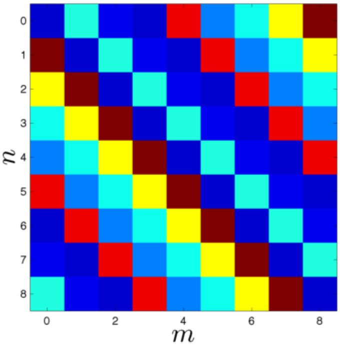
- All of the entries in a circulent matrix can be expressed in terms of the entries of the
  - 0-**th column**: $h[n] = h_{n,0}$ (this is a signal/column vector; call it h)
  - **Circularly time-reversed** 0**-th row**: $h[m] = h_{0,-m}$

- Example: Circulent matrix

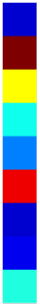
$$[\mathbf{H}]_{n,m} = h_{n,m} = h[(n-m)_N]$$

- Note the diagonals and circulent shifts!

$\mathbf{H} =$



$h =$



Summary

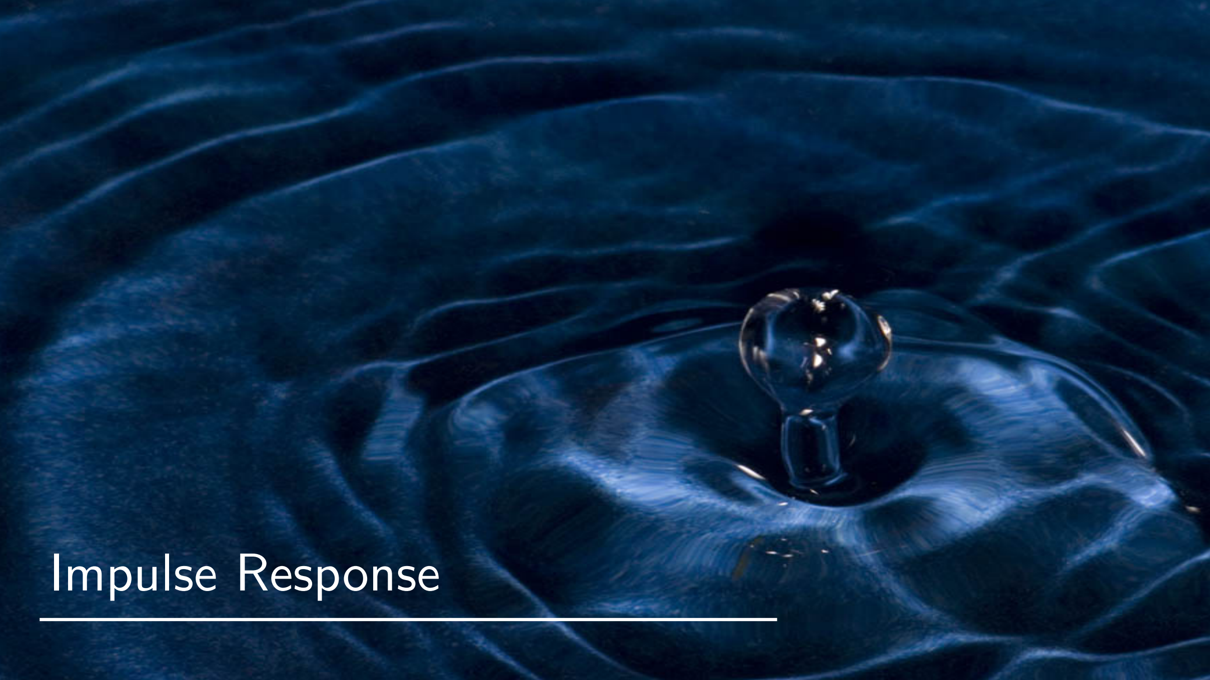
- LTI = Linear + Time-Invariant

- Fundamental signal processing system (and our focus for the rest of the course)

- Infinite-length signals: System = Toeplitz matrix $\mathbf{H}$
  - $[\mathbf{H}]_{n,m} \; = \; h_{n,m} \; = \; h[n-m]$

- Finite-length signals: System = Circulent matrix $\mathbf{H}$
  - $[\mathbf{H}]_{n,m} \; = \; h_{n,m} \; = \; h[(n-m)_N]$

Impulse Response

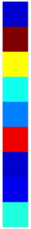


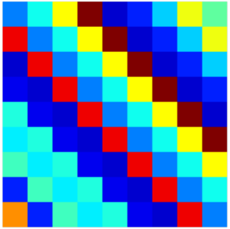
Recall: LTI Systems are Toeplitz Matrices (Infinite-Length Signals)

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

- LTI system $=$ multiplication by infinitely large Toeplitz matrix $\mathbf{H}$: $\quad y \;=\; \mathbf{H}x$
- All of the entries in $\mathbf{H}$ can be obtained from the
  - **0-th column**: $\quad h[n] = h_{n,0}$ $\quad$ (this is a signal/column vector; call it h)
  - **Time-reversed 0-th row**: $\quad h[m] = h_{0,-m}$
- $[\mathbf{H}]_{n,m} \;=\; h_{n,m}$

  $\qquad\qquad = \; h[n-m]$
- Columns/rows of $\mathbf{H}$ are shifted versions of the 0-th column/row

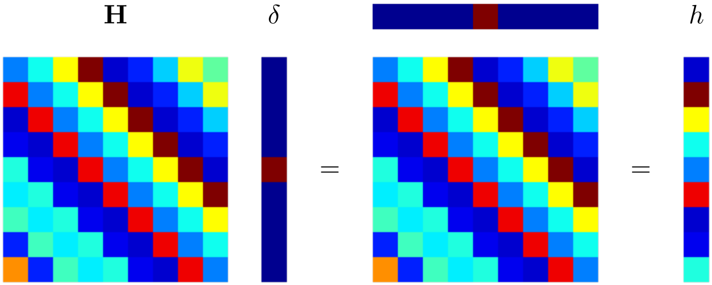
$h =$ 

$\mathbf{H} =$ 

Impulse Response (Infinite-Length Signals)

- The 0-th column of the matrix $\mathbf{H}$ – the column vector h – has a special interpretation

- Compute the output when the input is a **delta function** (impulse): $\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$



- This suggests that we call h the **impulse response** of the system

Impulse Response from Formulas (Infinite-Length Signals)

- General formula for LTI matrix multiplication

$$y[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\,x[m]$$

- Let the input $x[n] = \delta[n]$ and compute

$$\sum_{m=-\infty}^{\infty} h[n-m]\,\delta[m] \;=\; h[n] \quad \checkmark$$

$$\delta \longrightarrow \boxed{\;\mathcal{H}\;} \longrightarrow h$$

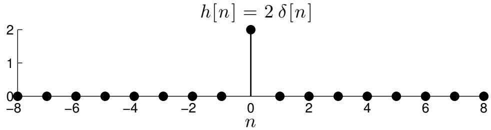
- The impulse response characterizes an LTI system
(that is, carries all of the information contained in the matrix $\mathbf{H}$)

$$x \longrightarrow \boxed{\;h\;} \longrightarrow y$$

Example: Impulse Response of the Scaling System

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

- Consider system for infinite-length signals; finite-length signal case is similar

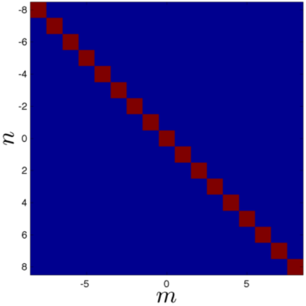


- Scaling system: $\quad y[n] \;=\; \mathcal{H}\{x[n]\} \;=\; 2\,x[n]$

- Impulse response: $\quad h[n] \;=\; \mathcal{H}\{\delta[n]\} \;=\; 2\,\delta[n]$

- Toeplitz system matrix:

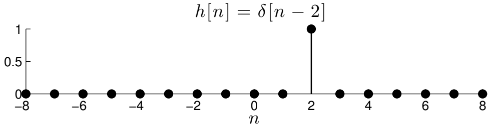
$$[\mathbf{H}]_{n,m} = h[n-m] = 2\,\delta[n-m]$$

$\mathbf{H} =$ 

Example: Impulse Response of the Shift System

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

- Consider system for infinite-length signals; finite-length signal case uses circular shift



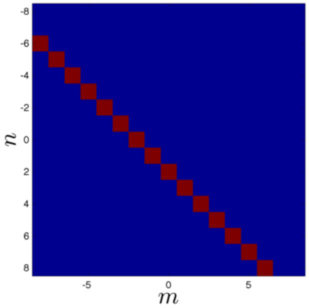
- Scaling system: $y[n] \;=\; \mathcal{H}\{x[n]\} \;=\; x[n-2]$

- Impulse response: $h[n] \;=\; \mathcal{H}\{\delta[n]\} \;=\; \delta[n-2]$

- Toeplitz system matrix:

$$[\mathbf{H}]_{n,m} = h[n-m] = \delta[n-m-2]$$

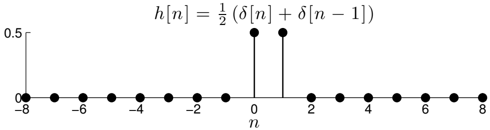
$\mathbf{H} =$



Example: Impulse Response of the Moving Average System

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

- Consider system for infinite-length signals; finite-length signal case is similar



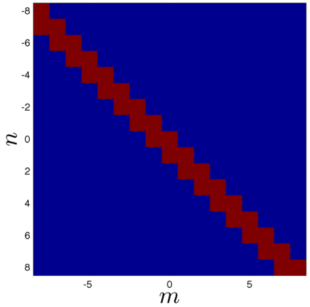
- Moving average system: $y[n] \;=\; \mathcal{H}\{x[n]\} \;=\; \frac{1}{2}\left(x[n] + x[n-1]\right)$

- Impulse response: $h[n] \;=\; \mathcal{H}\{\delta[n]\} \;=\; \frac{1}{2}\left(\delta[n] + \delta[n-1]\right)$

- Toeplitz system matrix:

$$[\mathbf{H}]_{n,m} = h[n-m] = \frac{1}{2}\left(\delta[n-m] + \delta[n-m-1]\right)$$

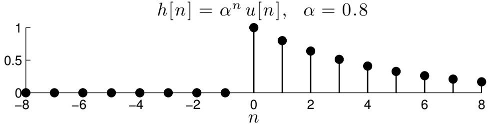
$\mathbf{H} \;=$



Example: Impulse Response of the Recursive Average System

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

- Consider system for infinite-length signals; finite-length signal case is similar



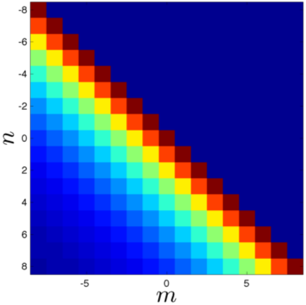
- Recursive average system: $\quad y[n] \;=\; \mathcal{H}\{x[n]\} \;=\; x[n] + \alpha\, y[n-1]$

- Impulse response: $\quad h[n] \;=\; \mathcal{H}\{\delta[n]\} \;=\; \alpha^n\, u[n]$

- Toeplitz system matrix:

$$[\mathbf{H}]_{n,m} = h[n-m] = \alpha^{n-m}\, u[n-m]$$

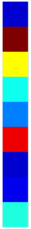
$\mathbf{H} \;=$

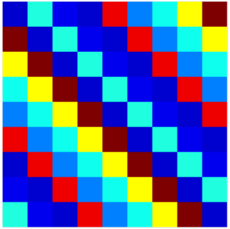


Recall: LTI Systems are Circulent Matrices (Finite-Length Signals)

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

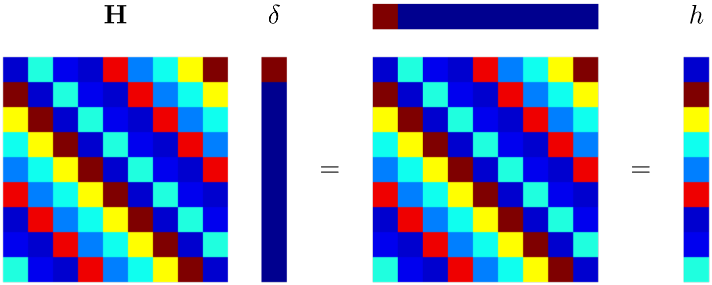
- LTI system = multiplication by $N \times N$ circulent matrix $\mathbf{H}$: $y = \mathbf{H}x$
- All of the entries in $\mathbf{H}$ can be obtained from the
  - **0-th column**: $h[n] = h_{n,0}$ (this is a signal/column vector; call it h)
  - **Time-reversed 0-th row**: $h[m] = h_{0,(-m)_N}$
- $[\mathbf{H}]_{n,m} = h_{n,m}$
  $= h[(n-m)_N]$
- Columns/rows of $\mathbf{H}$ are circularly shifted versions of the 0-th column/row

$h =$ 

$\mathbf{H} =$ 

Impulse Response (Finite-Length Signals)

- The 0-th column of the matrix $\mathbf{H}$ – the column vector h – has a special interpretation
- Compute the output when the input is a **delta function** (impulse): $\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$



- This suggests that we call h the **impulse response** of the system

Impulse Response from Formulas (Finite-Length Signals)

- General formula for LTI matrix multiplication

$$y[n] \;=\; \sum_{m=0}^{N-1} h[(n-m)_N]\, x[m]$$

- Let the input $x[n] = \delta[n]$ and compute

$$\sum_{m=0}^{N-1} h[(n-m)_N]\, \delta[m] \;=\; h[n] \quad \checkmark$$

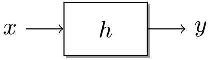
$$\delta \longrightarrow \boxed{\;\mathcal{H}\;} \longrightarrow h$$

- The impulse response characterizes an LTI system
(that is, carries all of the information contained in the matrix $\mathbf{H}$)

$$x \longrightarrow \boxed{\;h\;} \longrightarrow y$$

Summary

- LTI system $=$ multiplication by infinite-sized Toeplitz or $N \times N$ circulent matrix $\mathbf{H}$: $y \;=\; \mathbf{H}x$
- The **impulse response** h of an LTI system $=$ the response to an impulse δ
  - The impulse response is the 0-th column of the matrix $\mathbf{H}$
  - The impulse response characterizes an LTI system

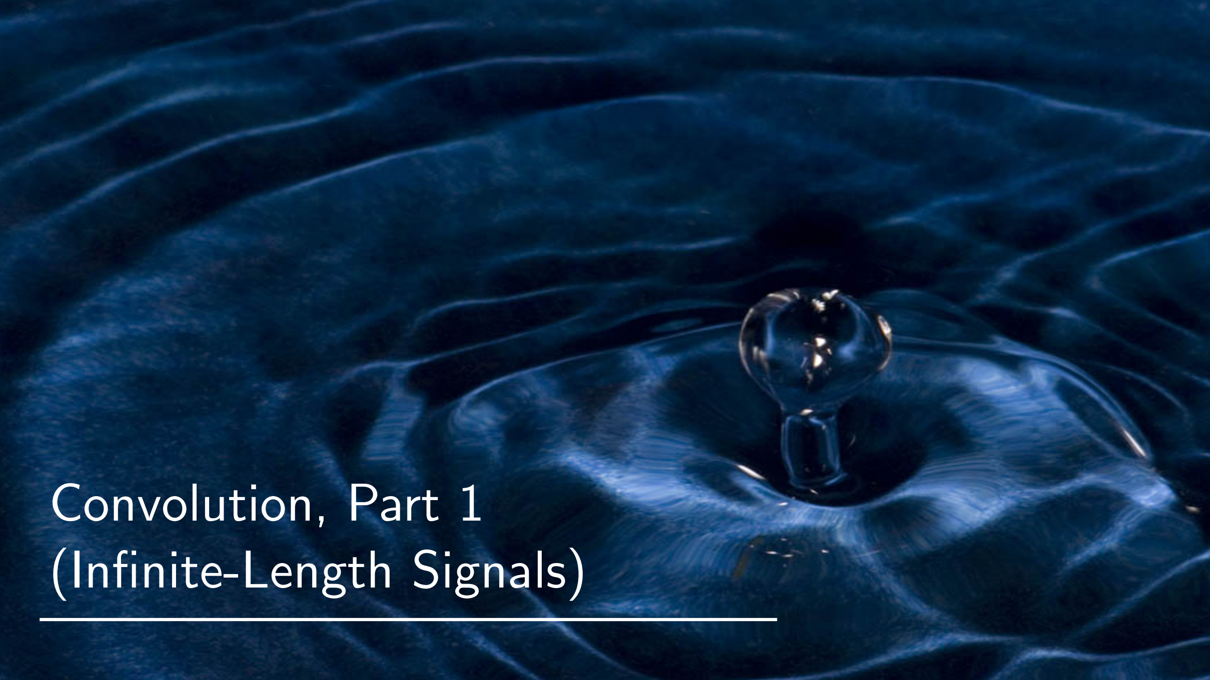


- Formula for the output signal y in terms of the input signal x and the impulse response h
  - Infinite-length signals

$$y[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m], \quad -\infty < n < \infty$$

  - Length-N signals

$$y[n] \;=\; \sum_{m=0}^{N-1} h[(n-m)_N]\, x[m], \quad 0 \le n \le N-1$$



Convolution, Part 1 (Infinite-Length Signals)

Three Ways to Compute the Output of an LTI System Given the Input

$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

1. If $\mathcal{H}$ is defined in terms of a formula or **algorithm**, apply the input x and compute $y[n]$ at each time point $n \in \mathbb{Z}$
   - This is how systems are usually applied in computer code and hardware

2. Find the impulse response h (by inputting $x[n] = \delta[n]$), form the **Toeplitz system matrix $\mathbf{H}$**, and multiply by the (infinite-length) input signal vector x to obtain $y = \mathbf{H}\, x$
   - This is not usually practical but is useful for conceptual purposes

3. Find the impulse response h and apply the formula for matrix/vector product for each $n \in \mathbb{Z}$

$$y[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m] \;=\; x[n] * h[n]$$

   - This is called **convolution** and is both conceptually and practically useful (Matlab command: `conv`)

Convolution as a Sequence of Inner Products

$$x \longrightarrow \boxed{h} \longrightarrow y$$

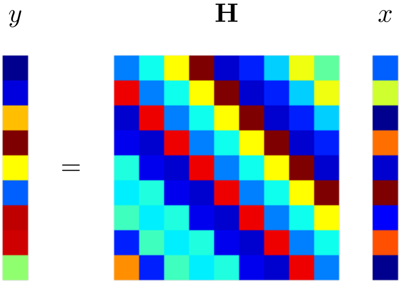
- Convolution formula

$$y[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

- To compute the entry $y[n]$ in the output vector y:

  1. **Time reverse** the impulse response vector h and **shift** it n time steps to the right (delay)
  2. Compute the **inner product** between the shifted impulse response and the input vector x

- Repeat for every n



A Seven-Step Program for Computing Convolution By Hand

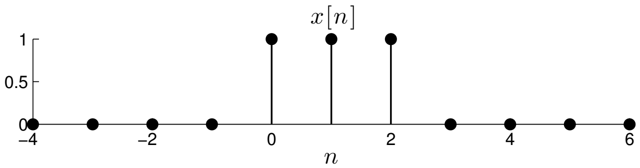
$$y[n] \;=\; x[n] * h[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

- **Step 1:** Decide which of x or h you will flip and shift; you have a choice since $x * h = h * x$
- **Step 2:** Plot $x[m]$ as a function of m
- **Step 3:** Plot the time-reversed impulse response $h[-m]$
- **Step 4:** To compute y at the time point n, plot the time-reversed impulse response after it has been shifted to the right (delayed) by n time units: $h[-(m-n)] = h[n-m]$
- **Step 5:** $y[n] =$ the inner product between the signals $x[m]$ and $h[n-m]$ (Note: for complex signals, do not complex conjugate the second signal in the inner product)
- **Step 6:** Repeat for all n of interest (potentially all $n \in \mathbb{Z}$)
- **Step 7:** Plot $y[n]$ and perform a reality check to make sure your answer seems reasonable

First Convolution Example (1)

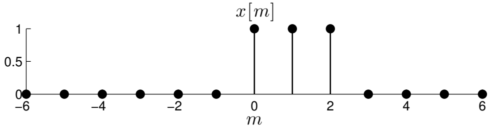
$$y[n] \;=\; x[n] * h[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

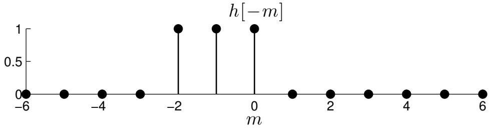
- Convolve a unit pulse with itself

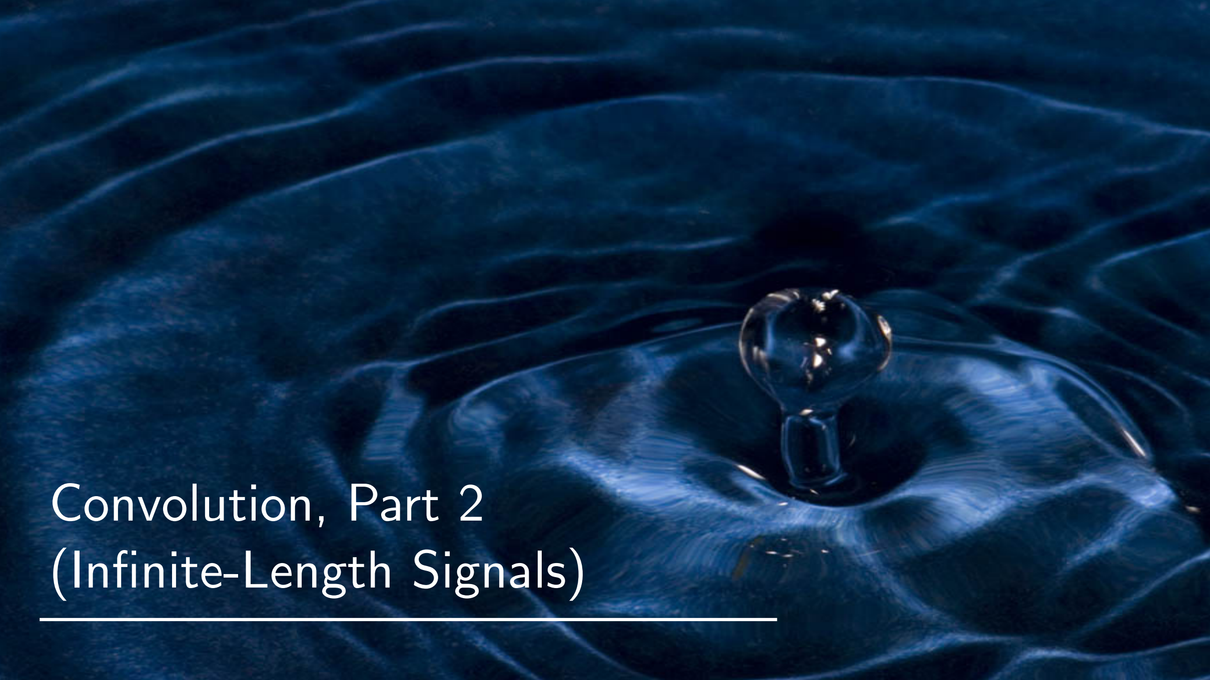


First Convolution Example (2)

$$y[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$







Convolution, Part 2 (Infinite-Length Signals)

A Seven-Step Program for Computing Convolution By Hand

$$y[n] \; = \; x[n] * h[n] \; = \; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

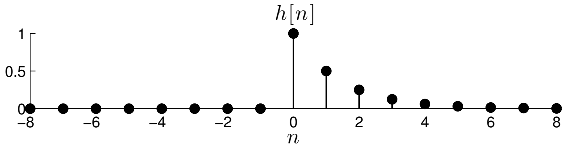
- **Step 1:** Decide which of x or h you will flip and shift; you have a choice since $x * h = h * x$
- **Step 2:** Plot $x[m]$ as a function of m
- **Step 3:** Plot the time-reversed impulse response $h[-m]$
- **Step 4:** To compute y at the time point n, plot the time-reversed impulse response after it has been shifted to the right (delayed) by n time units: $h[-(m-n)] = h[n-m]$
- **Step 5:** $y[n] =$ the inner product between the signals $x[m]$ and $h[n-m]$
(Note: for complex signals, do not complex conjugate the second signal in the inner product)
- **Step 6:** Repeat for all n of interest (potentially all $n \in \mathbb{Z}$)
- **Step 7:** Plot $y[n]$ and perform a reality check to make sure your answer seems reasonable

Second Convolution Example (1)

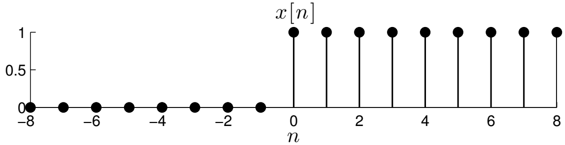
- Recall the **recursive average system**

$$y[n] = x[n] + \frac{1}{2}\,y[n-1]$$

and its impulse response $h[n] = \left(\frac{1}{2}\right)^n\,u[n]$

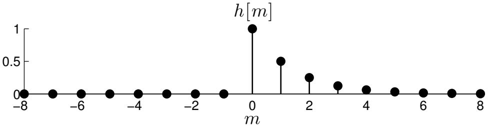


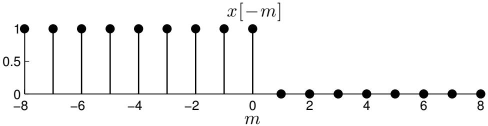
- Compute the output y when the input is a unit step $x[n] = u[n]$



Second Convolution Example (2)

$$y[n] \;=\; h[n] * x[n] \;=\; \sum_{m=-\infty}^{\infty} h[m]\, x[n-m]$$



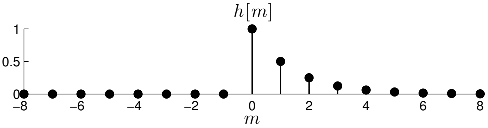


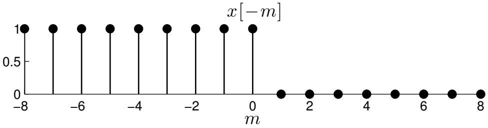
- Recall the super useful formula for the **finite geometric series**

$$\sum_{k=N_1}^{N_2} a^k \;=\; \frac{a^{N_1} - a^{N_2+1}}{1-a}, \quad N_1 \leq N_2$$

Second Convolution Example (3)

$$y[n] \;=\; h[n] * x[n] \;=\; \sum_{m=-\infty}^{\infty} h[m]\, x[n-m]$$





Summary

- **Convolution** formula for the output y of an LTI system given the input x and the impulse response h (infinite-length signals)

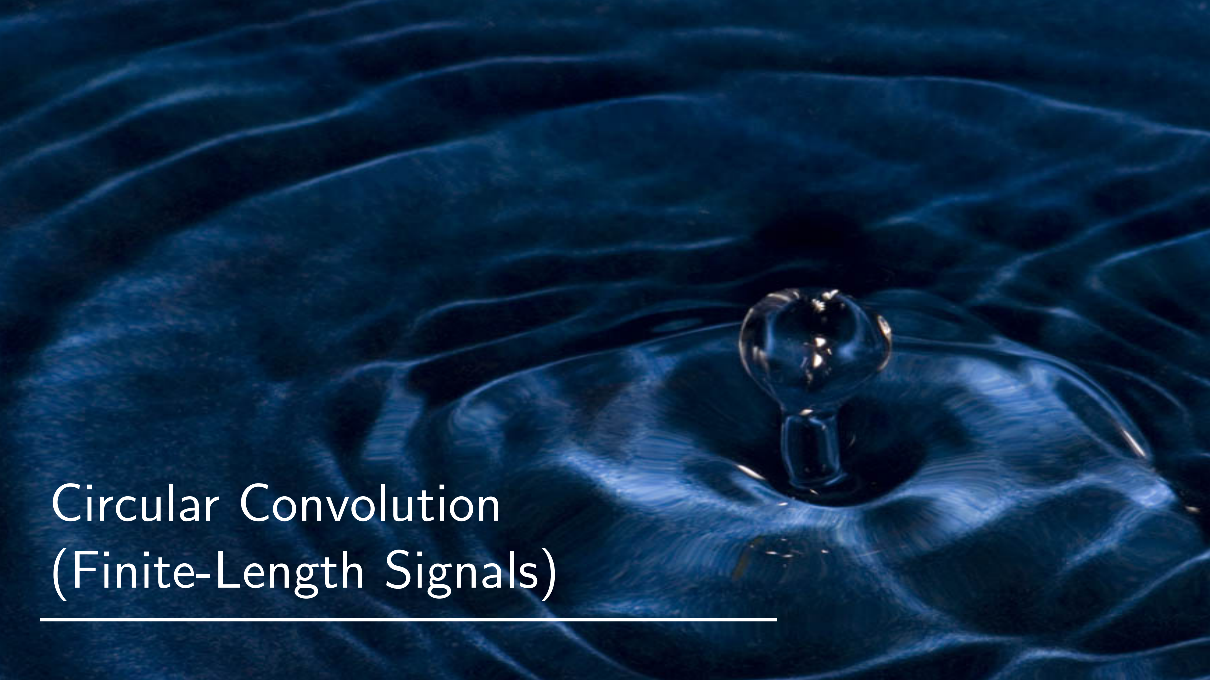
$$y[n] \;=\; x[n] * h[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

- Convolution is a sequence of inner products between the signal and the shifted, time-reversed impulse response

- Seven-step program for computing convolution by hand

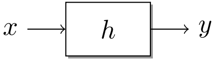
- Check your work and compute large convolutions using Matlab command `conv`

- Practice makes perfect!



Circular Convolution (Finite-Length Signals)

Circular Convolution as a Sequence of Inner Products



- Convolution formula

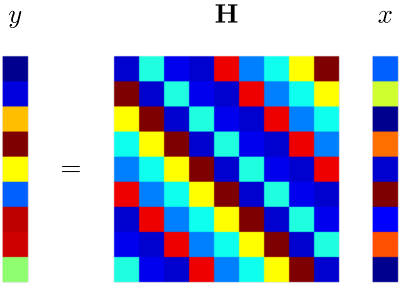
$$y[n] \;=\; x[n] \circledast h[n] \;=\; \sum_{m=0}^{N-1} h[(n-m)_N]\, x[m]$$

- To compute the entry $y[n]$ in the output vector y:

  1. **Circularly time reverse** the impulse response vector h and **circularly shift** it n time steps to the right (delay)

  2. Compute the **inner product** between the shifted impulse response and the input vector x

- Repeat for every n



A Seven-Step Program for Computing Circular Convolution By Hand

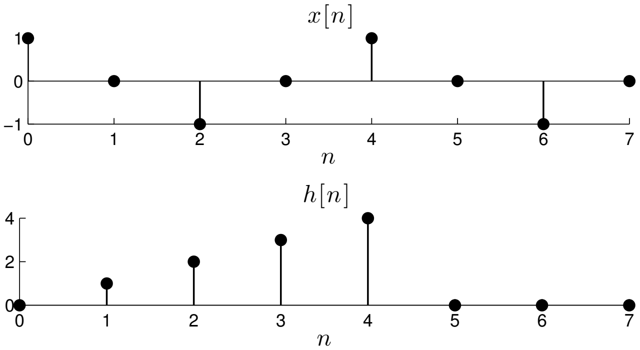
$$y[n] \;=\; x[n] \circledast h[n] \;=\; \sum_{m=0}^{N-1} h[(n-m)_N]\, x[m]$$

- **Step 1:** Decide which of x or h you will flip and shift; you have a choice since $x * h = h * x$
- **Step 2:** Plot $x[m]$ as a function of m on a clock with N "hours"
- **Step 3:** Plot the circularly time-reversed impulse response $h[(-m)_N]$ on a clock with N "hours"
- **Step 4:** To compute y at the time point n, plot the time-reversed impulse response after it has been shifted counter-clockwise (delayed) by n time units: $h[(-(m-n))_N] = h[(n-m)_N]$
- **Step 5:** $y[n] =$ the inner product between the signals $x[m]$ and $h[(n-m)_N]$
  (Note: for complex signals, do not complex conjugate the second signal in the inner product)
- **Step 6:** Repeat for all $n = 0, 1, \ldots, N-1$
- **Step 7:** Plot $y[n]$ and perform a reality check to make sure your answer seems reasonable

Circular Convolution Example (1)

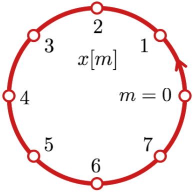
$$y[n] \;=\; x[n] \circledast h[n] \;=\; \sum_{m=0}^{N-1} h[(n-m)_N]\, x[m]$$

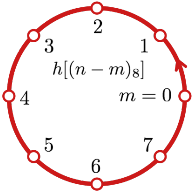
- For $N = 8$, circularly convolve a sinusoid x and a ramp h

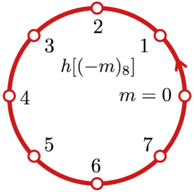


Circular Convolution Example (2)

$$y[n] \;=\; x[n] \circledast h[n] \;=\; \sum_{m=0}^{N-1} h[(n-m)_N]\, x[m]$$







Summary

- **Circular convolution** formula for the output y of an LTI system given the input x and the impulse response h (length-N signals)

$$y[n] \ = \ x[n] \circledast h[n] \ = \ \sum_{m=0}^{N-1} h[(n-m)_N] \, x[m]$$

- Circular convolution is a sequence of inner products between the signal and the circularly shifted, time-reversed impulse response

- Seven-step program for computing circular convolution by hand

- Check your work and compute large circular convolutions using Matlab command `cconv`

- Practice makes perfect!

Properties of Convolution

Properties of Convolution

$$x \longrightarrow \boxed{h} \longrightarrow y$$

- Input signal x, LTI system impulse response h, and output signal y are related by the **convolution**
  - Infinite-length signals

$$y[n] \;=\; x[n] * h[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m], \quad -\infty < n < \infty$$

  - Length-N signals

$$y[n] \;=\; x[n] \circledast h[n] \;=\; \sum_{m=0}^{N-1} h[(n-m)_N]\, x[m], \quad 0 \leq n \leq N-1$$

- Thanks to the Toeplitz/circulent structure of LTI systems, convolution has very special properties
- We will emphasize infinite-length convolution, but similar arguments hold for circular convolution except where noted

Convolution is Commutative

- **Fact:** Convolution is commutative: $x * h \;=\; h * x$

- These block diagrams are equivalent: $x \longrightarrow \boxed{\;h\;} \longrightarrow y \qquad h \longrightarrow \boxed{\;x\;} \longrightarrow y$

- Enables us to pick either h or x to flip and shift (or stack into a matrix) when convolving

- To prove, start with the convolution formula

$$y[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\,x[m] \;=\; x[n] * h[n]$$

and change variables to $k = n - m \;\Rightarrow\; m = n - k$

$$y[n] \;=\; \sum_{k=-\infty}^{\infty} h[k]\,x[n-k] \;=\; h[n] * x[n] \quad \checkmark$$

Cascade Connection of LTI Systems

- Impulse response of the **cascade** (aka series connection) of two LTI systems: $y = \mathbf{H}_1 \mathbf{H}_2\, x$

$$x \longrightarrow \boxed{h_1} \longrightarrow \boxed{h_2} \longrightarrow y$$

$$x \longrightarrow \boxed{h_1 * h_2} \longrightarrow y$$

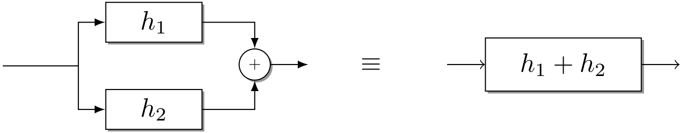
- Interpretation: The product of two Toeplitz/circulent matrices is a Toeplitz/circulent matrix

- Easy proof by picture; find impulse response the old school way

$$\delta \longrightarrow \boxed{h_1} \longrightarrow h_1 \qquad \longrightarrow \boxed{h_2} \longrightarrow h_1 * h_2$$

Parallel Connection of LTI Systems

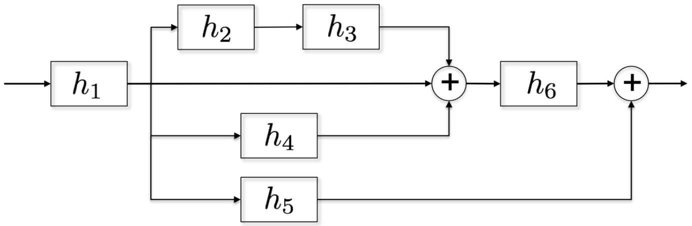
- Impulse response of the **parallel connection** of two LTI systems $y = (\mathbf{H}_1 + \mathbf{H}_2)\, x$

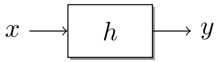


- Proof is an easy application of the linearity of an LTI system

Example: Impulse Response of a Complicated Connection of LTI Systems

- Compute the overall effective impulse response of the following system



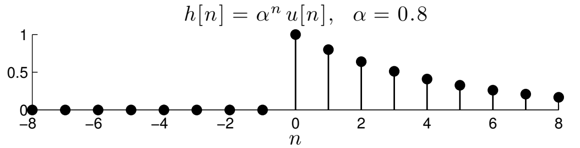


Causal Systems

DEFINITION

A system $\mathcal{H}$ is **causal** if the output $y[n]$ at time n depends only the input $x[m]$ for times $m \leq n$. In words, causal systems do not look into the future

- **Fact:** An LTI system is causal if its impulse response is causal: $h[n] = 0$ for $n < 0$



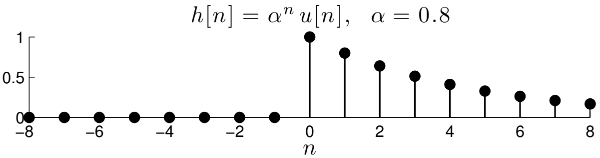
- To prove, note that the convolution

$$y[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

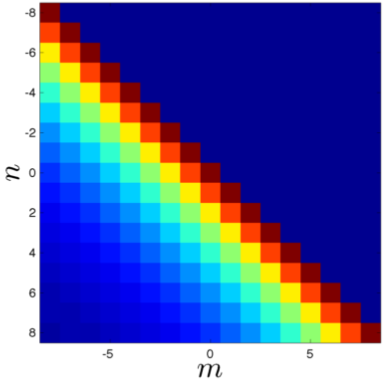
does not look into the future if $h[n-m] = 0$ when $m > n$; equivalently, $h[n'] = 0$ when $n' < 0$

Causal System Matrix

- **Fact:** An LTI system is causal if its impulse response is causal: $h[n] = 0$ for $n < 0$



- Toeplitz system matrix is **lower triangular**

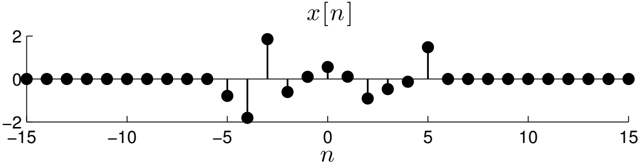


Duration of Convolution

DEFINITION

The signal x has **support interval** $[N_1, N_2]$, $N_1 \leq N_2$, if $x[n] = 0$ for all $n < N_1$ and $n > N_2$. The **duration** D_x of x equals $N_2 - N_1 + 1$

- Example: A signal with support interval $[-5, 5]$ and duration 11 samples



- **Fact:** If x has duration D_x samples and h has duration D_h samples, then the convolution $y = x * h$ has duration at most $D_x + D_h - 1$ samples (proof by picture is simple)

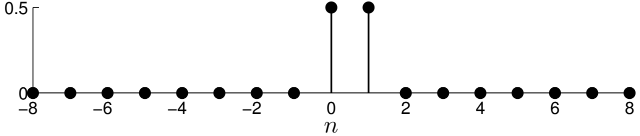
Duration of Impulse Response – FIR

DEFINITION

An LTI system has a **finite impulse response** (FIR) if the duration of its impulse response h is finite

- Example: Moving average system $\quad y[n] \;=\; \mathcal{H}\{x[n]\} \;=\; \frac{1}{2}\left(x[n] + x[n-1]\right)$

$$h[n] = \tfrac{1}{2}\left(\delta[n] + \delta[n-1]\right)$$

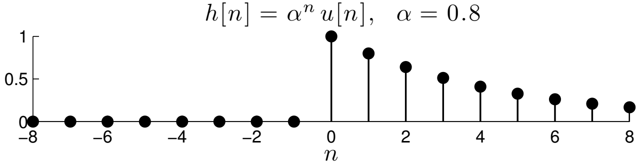


Duration of Impulse Response – IIR

DEFINITION

An LTI system has an **infinite impulse response** (IIR) if the duration of its impulse response h is infinite

- Example: Recursive average system $\quad y[n] \;=\; \mathcal{H}\{x[n]\} \;=\; x[n] + \alpha\, y[n-1]$



- Note: Obviously the FIR/IIR distinction applies only to infinite-length signals

Implementing Infinite-Length Convolution with Circular Convolution

- Consider two infinite-length signals: x has duration D_x samples and h has duration D_h samples, $D_x, D_h < \infty$

- Recall that their infinite-length convolution $y = x * h$ has duration at most $D_x + D_h - 1$ samples

- Armed with this fact, we can implement infinite-length convolution using circular convolution
  1. Extract the D_x-sample support interval of x and zero pad so that the resulting signal x' is of length $D_x + D_h - 1$
  2. Perform the same operations on h to obtain h'
  3. Circularly convolve $x' \circledast h'$ to obtain y'

- **Fact:** The values of the signal y' will coincide with those of the infinite-length convolution $y = x * h$ within its support interval

- How does it work? The zero padding effectively converts circular shifts (finite-length signals) into regular shifts (infinite-length signals) (Easy to try out in Matlab!)

Summary

- Convolution has very special and beautiful properties
- Convolution is commutative
- Convolutions (LTI systems) can be connected in cascade and parallel
- An LTI system is causal if its impulse response is causal
- LTI systems are either FIR or IIR
- Can implement infinite-length convolution using circular convolution when the signals have finite duration (important later for "fast convolution" using the FFT)

Convolution Examples in Matlab

Convolution in Matlab

- You can build your intuition and solve real-world problems using Matlab's convolution functions

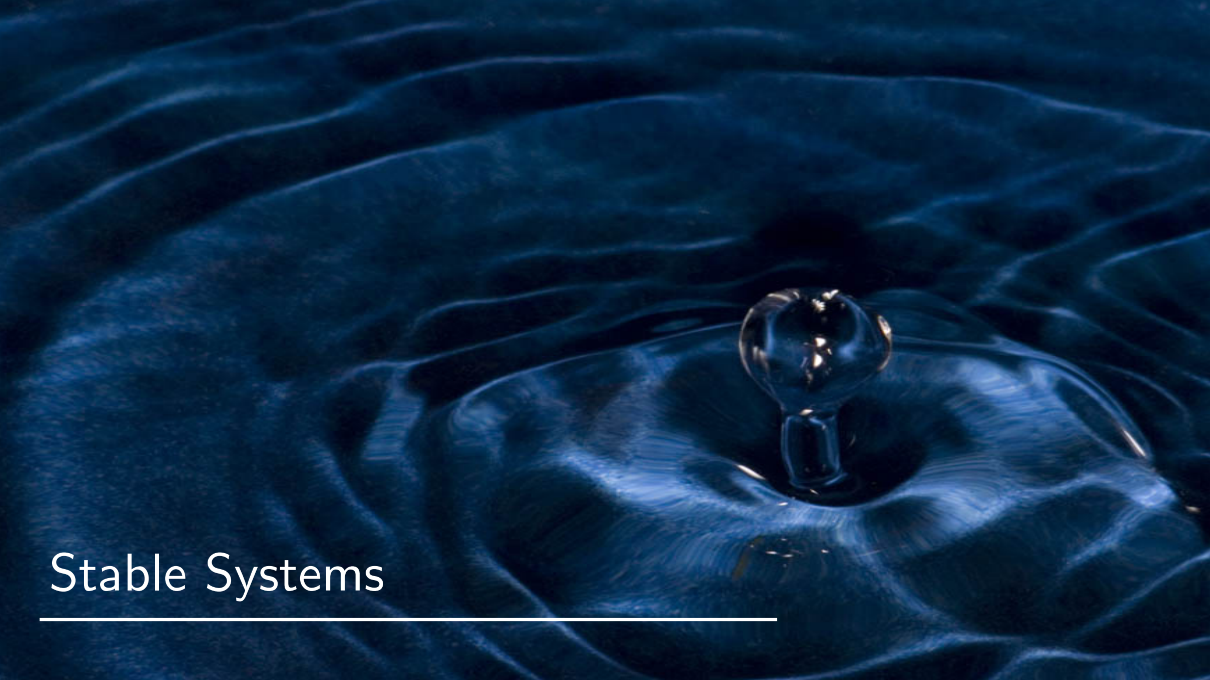
- Matlab's `conv` command implements **infinite-length convolution**

$$y[n] \;=\; x[n] * h[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

  by implicitly infinitely zero-padding the signal vectors; signal lengths need not be the same

- Matlab's `cconv` command implements length-N **circular convolution**

$$y[n] \;=\; x[n] \circledast h[n] \;=\; \sum_{m=0}^{N-1} h[(n-m)_N]\, x[m]$$



Stable Systems

Stable Systems (1)

- With a **stable** system, a “well-behaved” input always produces a “well-behaved” output

$$\text{“well-behaved” } x \longrightarrow \boxed{\;h\;} \longrightarrow \text{“well-behaved” } y$$

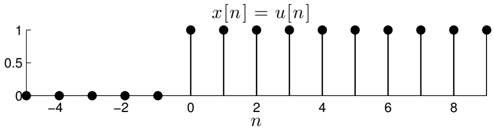
- Stability is essential to ensuring the proper and safe operation of myriad systems
  - Steering systems
  - Braking systems
  - Robotic navigation
  - Modern aircraft
  - International Space Station
  - Internet IP packet communication (TCP) . . .

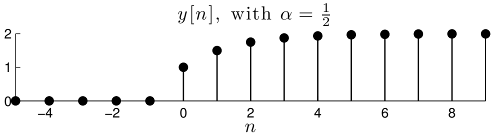
Stable Systems (2)

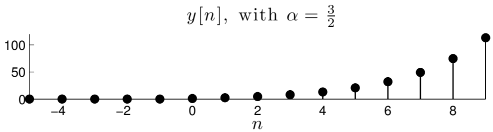
- With a **stable** system, a “well-behaved” input always produces a “well-behaved” output

“well-behaved” $x \longrightarrow \boxed{h} \longrightarrow$ “well-behaved” y

- Example: Recall the recursive average system $\quad y[n] \; = \; \mathcal{H}\{x[n]\} \; = \; x[n] + \alpha\, y[n-1]$
  Consider a step function input $\quad x[n] \; = \; u[n]$







Well-Behaved Signals

- With a **stable** system, a "well-behaved" input always produces a "well-behaved" output

$$\text{"well-behaved" } x \longrightarrow \boxed{h} \longrightarrow \text{"well-behaved" } y$$

- How to measure how "well-behaved" a signal is? Different measures give different notions of stability

- One reasonable measure: A signal x is well behaved if it is **bounded** (recall that $\sup$ is like $\max$)

$$\|x\|_\infty \; = \; \sup_n |x[n]| \; < \; \infty$$

Bounded-Input Bounded-Output (BIBO) Stability

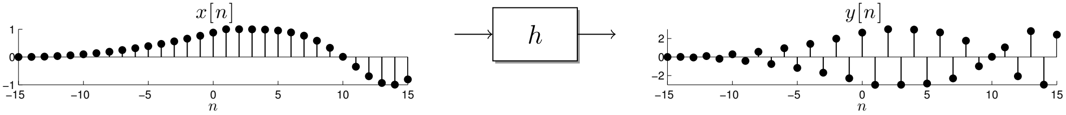
BIBO Stability (1)

DEFINITION

An LTI system is **bounded-input bounded-output (BIBO) stable** if a bounded input x always produces a bounded output y

$$\text{bounded } x \longrightarrow \boxed{h} \longrightarrow \text{bounded } y$$

- Bounded input and output means $\|x\|_\infty < \infty$ and $\|y\|_\infty < \infty$, or that there exist constants $A, C < \infty$ such that $|x[n]| < A$ and $|y[n]| < C$ for all n



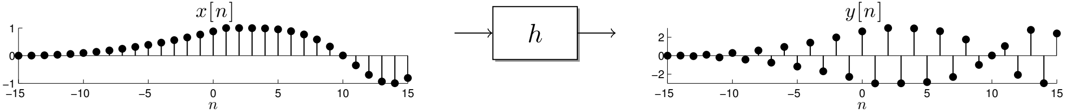
BIBO Stability (2)

DEFINITION

An LTI system is **bounded-input bounded-output (BIBO) stable** if a bounded input x always produces a bounded output y

$$\text{bounded } x \longrightarrow \boxed{h} \longrightarrow \text{bounded } y$$

- Bounded input and output means $\|x\|_\infty < \infty$ and $\|y\|_\infty < \infty$



- **Fact:** An LTI system with impulse response h is BIBO stable <u>if and only if</u>

$$\|h\|_1 = \sum_{n=-\infty}^{\infty} |h[n]| < \infty$$

BIBO Stability – Sufficient Condition

- Prove that if $\|h\|_1 < \infty$ then the system is BIBO stable – for any input $\|x\|_\infty < \infty$ the output $\|y\|_\infty < \infty$

- Recall that $\|x\|_\infty < \infty$ means there exist a constant A such that $|x[n]| < A < \infty$ for all n

- Let $\|h\|_1 = \sum_{n=-\infty}^{\infty} |h[n]| = B < \infty$

- Compute a bound on $|y[n]|$ using the convolution of x and h and the bounds A and B

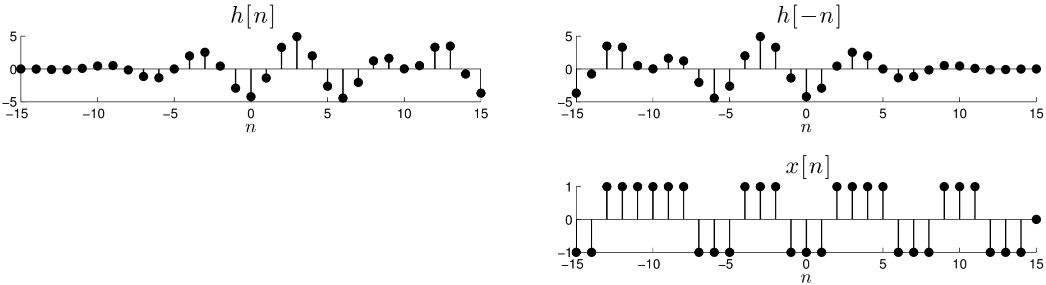
$$
\begin{aligned}
|y[n]| &= \left| \sum_{m=-\infty}^{\infty} h[n-m]\, x[m] \right| < \sum_{m=-\infty}^{\infty} |h[n-m]|\, |x[m]| \\
&< \sum_{m=-\infty}^{\infty} |h[n-m]|\, A = A \sum_{k=-\infty}^{\infty} |h[k]| = A\, B = C < \infty
\end{aligned}
$$

- Since $|y[n]| < C < \infty$ for all n, $\|y\|_\infty < \infty$ ✓

BIBO Stability – Necessary Condition (1)

- Prove that <u>if $\|h\|_1 = \infty$</u> then the system is <u>not</u> BIBO stable – there exists an input $\|x\|_\infty < \infty$ such that the output $\|y\|_\infty = \infty$
  - Assume that x and h are real-valued; the proof for complex-valued signals is nearly identical

- Given an impulse response h with $\|h\|_1 = \infty$ (assume complex-valued), form the tricky special signal $x[n] = \mathrm{sgn}(h[-n])$
  - $x[n]$ is the $\pm$ sign of the time-reversed impulse response $h[-n]$
  - Note that x is bounded: $|x[n]| \leq 1$ for all n



BIBO Stability – Necessary Condition (2)

- We are proving that that if $\|h\|_1 = \infty$ then the system is not BIBO stable – there exists an input $\|x\|_\infty < \infty$ such that the output $\|y\|_\infty = \infty$

- Armed with the tricky special signal x, compute the output $y[n]$ at the time point $n = 0$

$$\begin{aligned} y[0] &= \sum_{m=-\infty}^{\infty} h[0-m]\, x[m] = \sum_{m=-\infty}^{\infty} h[-m]\, \mathrm{sgn}(h[-m]) \\ &= \sum_{m=-\infty}^{\infty} |h[-m]| = \sum_{k=-\infty}^{\infty} |h[k]| = \infty \end{aligned}$$

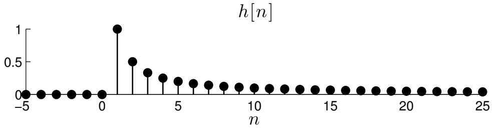
- So, even though x was bounded, y is not bounded; so system is not BIBO stable

BIBO System Examples (1)

- Absolute summability of the impulse response h determines whether an LTI systems is BIBO stable or not

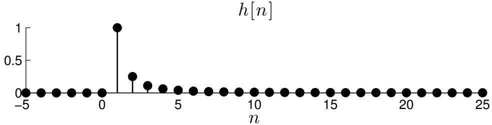
- Example: $h[n] = \begin{cases} \frac{1}{n} & n \geq 1 \\ 0 & \text{otherwise} \end{cases}$

  $\|h\|_1 = \sum_{n=1}^{\infty} \left|\frac{1}{n}\right| = \infty \;\Rightarrow\;$ not BIBO

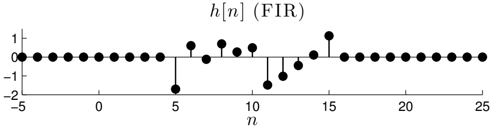


- Example: $h[n] = \begin{cases} \frac{1}{n^2} & n \geq 1 \\ 0 & \text{otherwise} \end{cases}$

  $\|h\|_1 = \sum_{n=1}^{\infty} \left|\frac{1}{n^2}\right| = \frac{\pi^2}{6} \;\Rightarrow\;$ BIBO



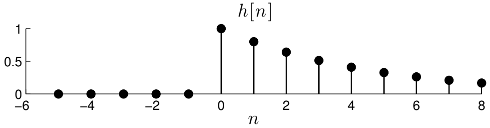
- Example: h FIR $\;\Rightarrow\;$ BIBO



BIBO System Examples (2)

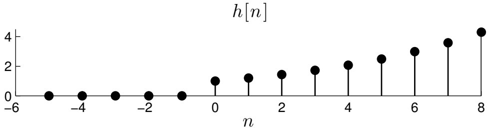
- Example: Recall the recursive average system $y[n] \;=\; \mathcal{H}\{x[n]\} \;=\; x[n] + \alpha\, y[n-1]$
- Impulse response: $h[n] \;=\; \alpha^n\, u[n]$
- For $|\alpha| < 1$

$$\|h\|_1 = \sum_{n=0}^{\infty} |\alpha|^n = \frac{1}{1-|\alpha|} < \infty \;\Rightarrow\; \text{BIBO}$$



- For $|\alpha| > 1$

$$\|h\|_1 = \sum_{n=0}^{\infty} |\alpha|^n = \infty \;\Rightarrow\; \text{not BIBO}$$



Summary

- Signal processing applications typically dictate that the system be **stable**, meaning that "well-behaved inputs" produce "well-behaved outputs"

- Measure "well-behavedness" of a signal using the ∞-norm (bounded signal)

- BIBO stability: bounded inputs always produce bounded outputs iff the impulse response h is such that $\|h\|_1 < \infty$

- When a system is not BIBO stable, all hope is not lost; unstable systems can often by **stabilized** using **feedback** (more on this later)

Acknowledgements