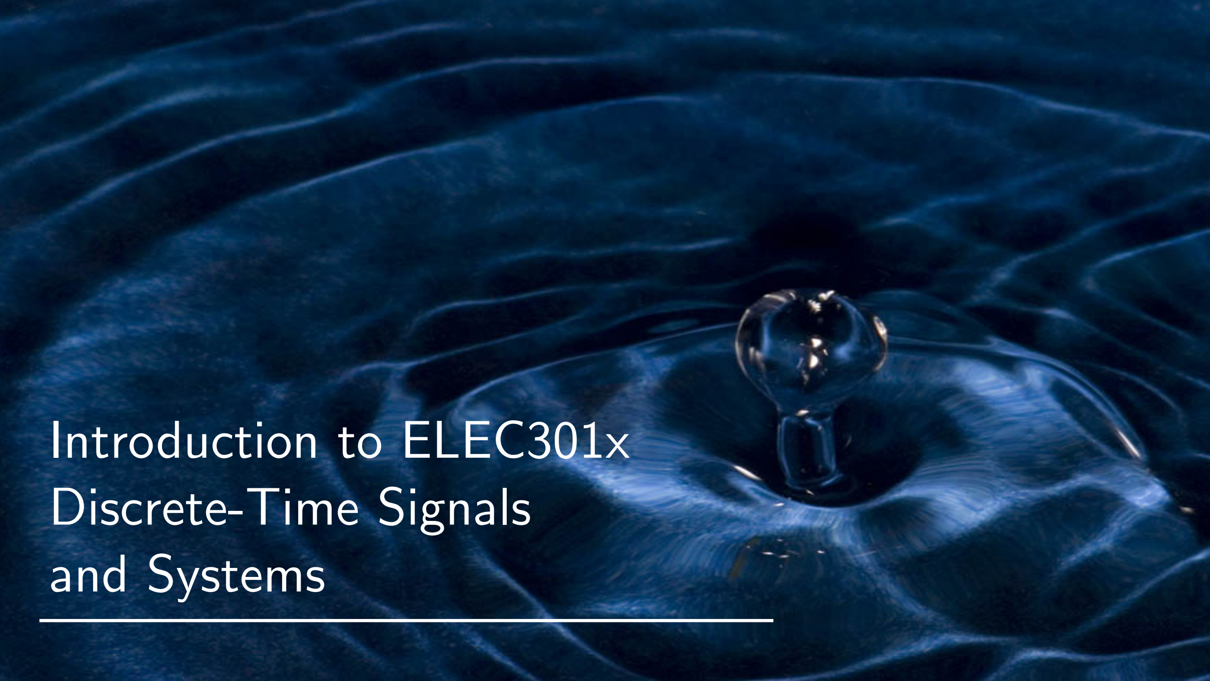


Introduction to ELEC301x Discrete-Time Signals and Systems

Welcome to Elec301x – Discrete Time Signals and Systems

- This is an introductory course on **signal processing**, which studies **signals** and **systems**

DEFINITION

Signal (n): A detectable physical quantity . . . by which messages or information can be transmitted (Merriam-Webster)

- Signals carry **information**
- Examples:
  - Speech signals transmit language via acoustic waves
  - Radar signals transmit the position and velocity of targets via electromagnetic waves
  - Electrophysiology signals transmit information about processes inside the body
  - Financial signals transmit information about events in the economy

Welcome to Elec301x – Discrete Time Signals and Systems

- **Systems** manipulate the information carried by signals

DEFINITION

Signal processing involves the theory and application of

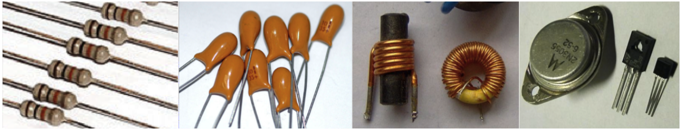
- filtering, coding, transmitting, estimating, detecting, analyzing, recognizing, synthesizing, recording, and reproducing signals by digital or analog devices or techniques
- where signal includes audio, video, speech, image, communication, geophysical, sonar, radar, medical, musical, and other signals

(IEEE Signal Processing Society Constitutional Amendment, 1994)

IEEE
Signal Processing Society®

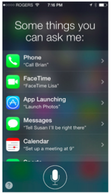
Signal Processing

- Signal processing has traditionally been a part of electrical and computer engineering
- But now expands into applied mathematics, statistics, computer science, geophysics, and host of application disciplines
- Initially **analog** signals and systems implemented using resistors, capacitors, inductors, and transistors



- Since the 1940s increasingly **digital** signals and systems implemented using computers and computer code (Matlab, Python, C, . . .)
  - Advantages of digital include stability and programmability
  - As computers have shrunk, digital signal processing has become ubiquitous

Digital Signal Processing Applications



ROGERS 7:16 PM

Some things you can ask me:

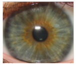
Phone
"Call Brian"

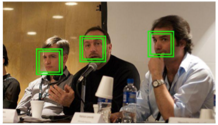
FaceTime
"FaceTime Lisa"

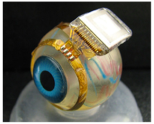
App Launching
"Launch Photos"

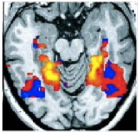
Messages
"Tell Susan I'll be right there"

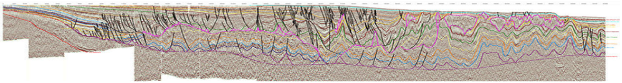
Calendar
"Set up a meeting at 9"



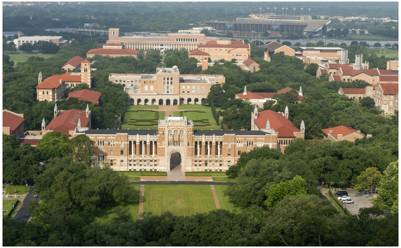








Rice ELEC301x



- This edX course consists of one-half of the core Electrical and Computer Engineering course entitled "Signals and Systems" taught at Rice University in Houston, Texas, USA (see www.dsp.rice.edu)
- Goals: Develop intuition into and learn how to reason analytically about signal processing problems
- Video lectures, primary sources, supplemental materials, practice exercises, homework, programming case studies, final exam
- Integrated Matlab!
- **Important:** This is a mathematical treatment of signals and systems (no pain, no gain!)

Before You Start

- Please make sure you have a solid understanding of
  - Complex numbers and arithmetic
  - Linear algebra (vectors, matrices, dot products, eigenvectors, bases . . .)
  - Series (finite and infinite)
  - Calculus of a single variable (derivatives and integrals)
  - Matlab
- To test your readiness or refresh your knowledge, visit the “Pre-class Mathematics Refresher” section of the course

Course Outline

- Week 1: Signals
- Week 2: Systems
- Week 3: Discrete Fourier Transform (DFT)
- Week 4: Discrete-Time Fourier Transform (DTFT)
- Week 5: z Transform
- Week 6: Filter Design
- Week 7: Study Week and Final Exam

What You Should Do Each Week

- Watch the Lecture videos
- Do the Exercises (on the page to the right of the videos)
- As necessary, refer to the lesson's Supplemental Resources (the page to the right of the exercises)
- Do the homework problems
- Some weeks will also have graded MATLAB case study homework problems

Logistics and Grading

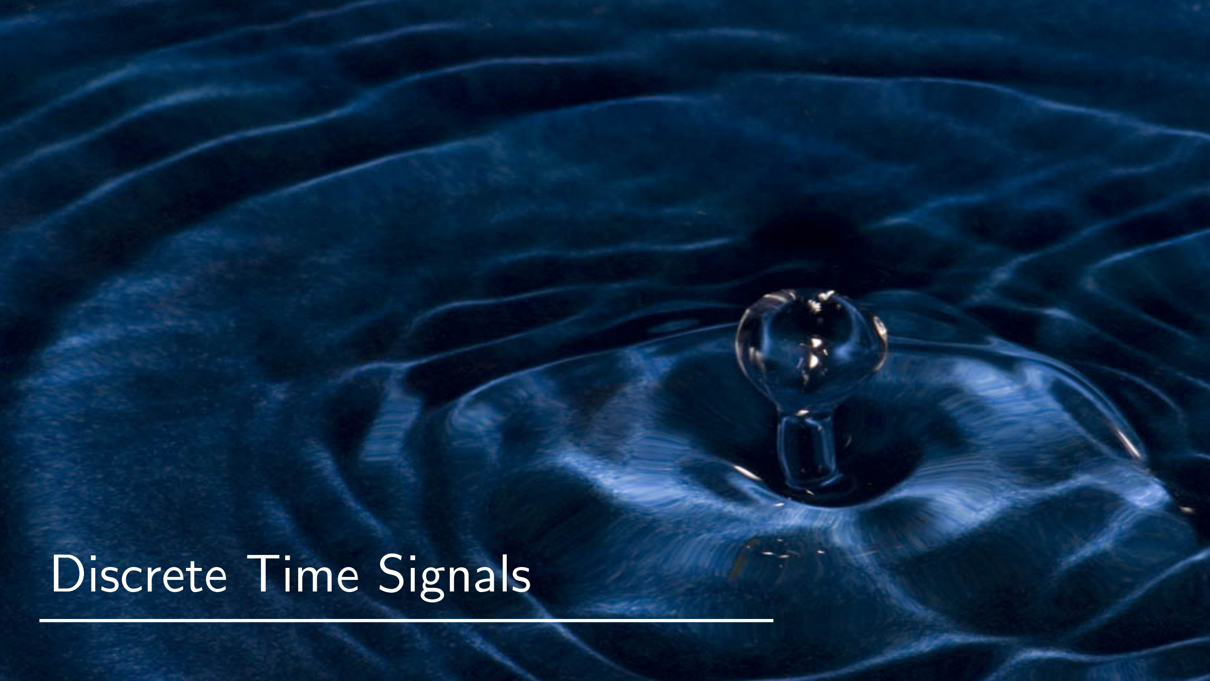
- How to get help: **Course Discussion** page
  - Use a thread set up for a particular topic, or
  - Start a new thread
- Rules for discussion
  - Be respectful and helpful
  - Do not reveal answers to any problem that will be graded
- Grading

Homework	22%	(lowest score dropped)
Practice exercises	15%	
Final exam	30%	
Matlab case studies (four)	15%	(lowest score dropped)
Exit survey	3%	

- Passing grade: 60%

Supplemental Materials

- After the video lecture and a practice exercise or two, you will often see additional **Supplemental Resources**
- Sometimes these will contain background material to provide motivation for the topic
- Sometimes these will provide a refresher of pre-requisite concepts
- Sometimes these will provide deeper explanations of the content (more rigorous proofs, etc.)
- Sometimes a particular signal processing application will be showcased
- **Important:** Though the content in these resources will not be assessed in the homework or exam, you may find that they help you to understand a concept better or increase your interest in it



Discrete Time Signals

Before We Start

- Please make sure you have a solid understanding of
  - Complex numbers and arithmetic
  - Linear algebra (vectors, matrices, dot products, eigenvectors, bases . . .)
  - Series (finite and infinite)
  - Calculus of a single variable (derivatives and integrals)
  - Matlab
- To test your readiness or refresh your knowledge, visit the “Pre-class Mathematics Refresher” section of the course

Signals

DEFINITION

Signal (n): A detectable physical quantity . . . by which messages or information can be transmitted (Merriam-Webster)

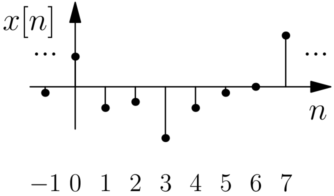
- Signals carry **information**
- Examples:
  - Speech signals transmit language via acoustic waves
  - Radar signals transmit the position and velocity of targets via electromagnetic waves
  - Electrophysiology signals transmit information about processes inside the body
  - Financial signals transmit information about events in the economy
- **Signal processing systems** manipulate the information carried by signals
- This is a course about signals and systems

Signals are Functions

DEFINITION

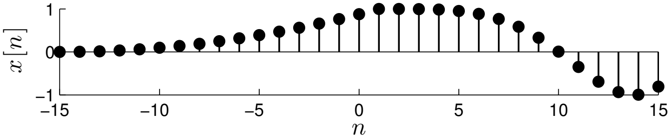
A **signal** is a function that maps an independent variable to a dependent variable.

- Signal $x[n]$: each value of n produces the value $x[n]$
- In this course, we will focus on **discrete-time** signals:
  - Independent variable is an **integer**: $n \in \mathbb{Z}$ (will refer to as time)
  - Dependent variable is a real or complex number: $x[n] \in \mathbb{R}$ or $\mathbb{C}$

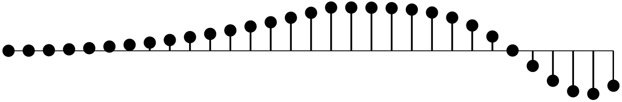


Plotting Real Signals

- When $x[n] \in \mathbb{R}$ (ex: temperature in a room at noon on Monday), we use one signal plot

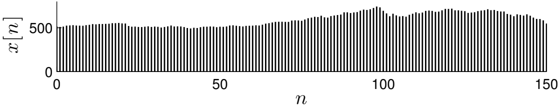


- When it is clear from context, we will often suppress the labels on one or both axes, like this

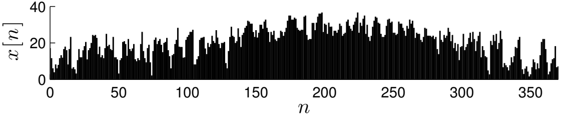


A Menagerie of Signals

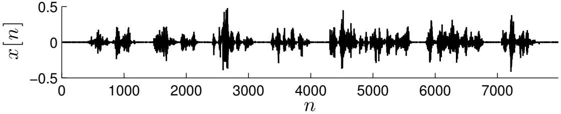
- Google Share daily share price for 5 months



- Temperature at Houston Intercontinental Airport in 2013 (Celcius)



- Excerpt from Shakespeare's *Hamlet*

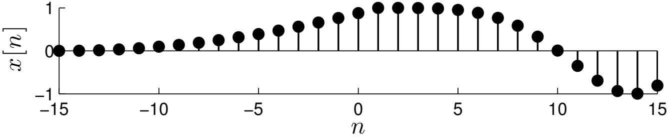


Plotting Signals Correctly

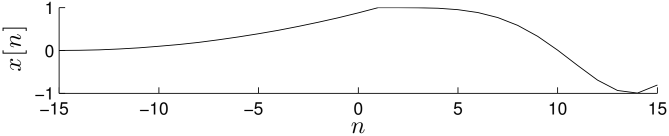
- In a discrete-time signal $x[n]$, the independent variable n is discrete (integer)

- To plot a discrete-time signal in a program like Matlab, you should use the `stem` or similar command and not the `plot` command

- Correct:



- Incorrect:

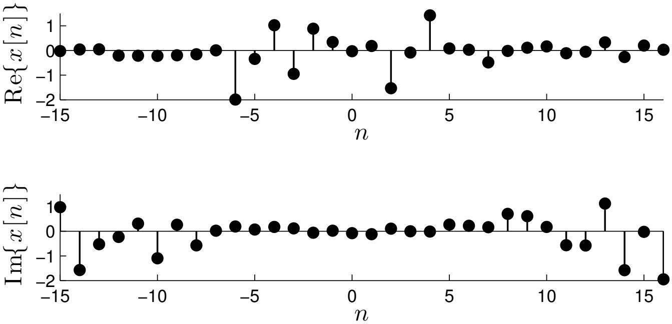


Plotting Complex Signals

- Recall that a complex number $a \in \mathbb{C}$ can be equivalently represented two ways:
  - Polar form: $a = |a|\, e^{j\angle a}$
  - Rectangular form: $a = \mathrm{Re}\{a\} + j\,\mathrm{Im}\{a\}$
- Here $j = \sqrt{-1}$ (engineering notation; mathematicians use $i = \sqrt{-1}$)
- When $x[n] \in \mathbb{C}$ (ex: magnitude and phase of an electromagnetic wave), we use two signal plots
  - Rectangular form: $x[n] = \mathrm{Re}\{x[n]\} + j\,\mathrm{Im}\{x[n]\}$
  - Polar form: $x[n] = |x[n]|\, e^{j\angle x[n]}$

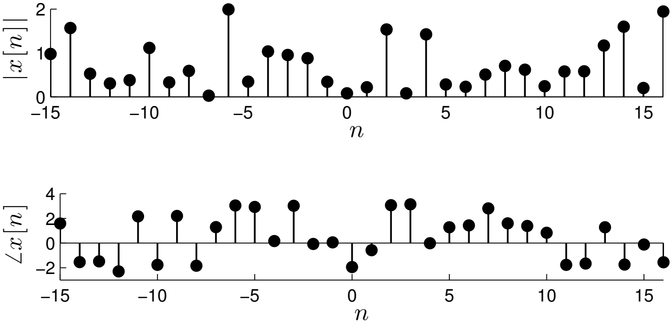
Plotting Complex Signals (Rectangular Form)

- Rectangular form: $x[n] = \mathrm{Re}\{x[n]\} + j\,\mathrm{Im}\{x[n]\} \in \mathbb{C}$



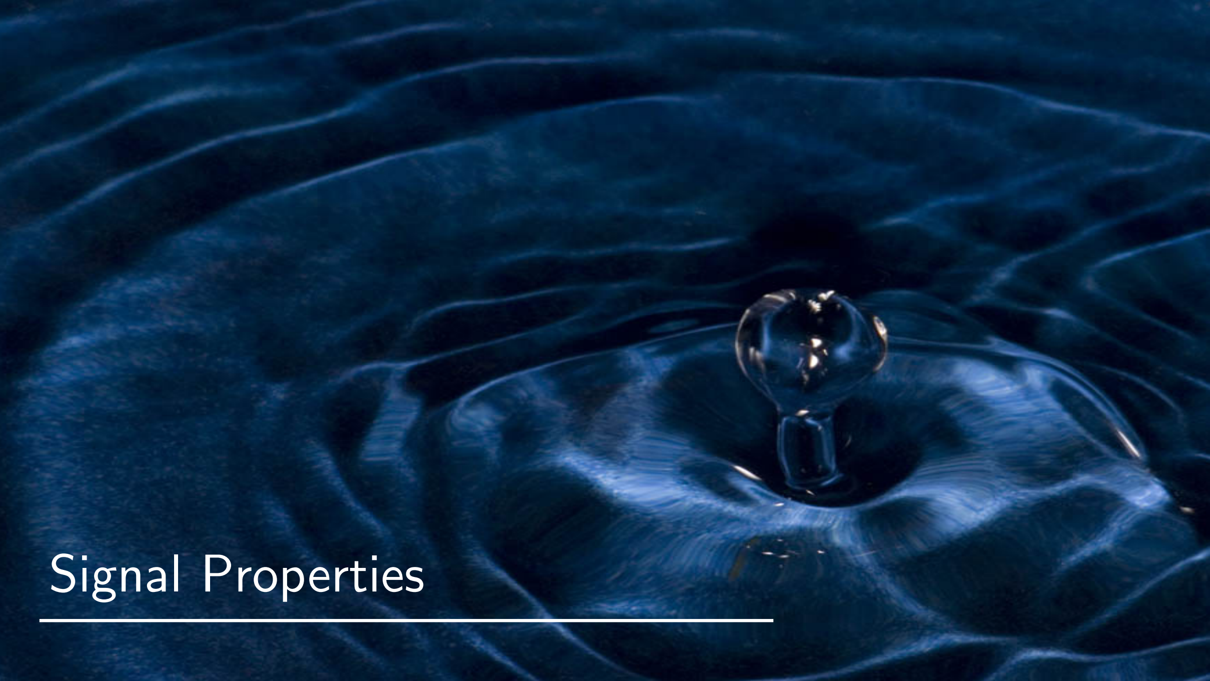
Plotting Complex Signals (Polar Form)

- Polar form: $x[n] = |x[n]|\, e^{j\angle(x[n])} \in \mathbb{C}$



Summary

- Discrete-time signals
  - Independent variable is an integer: $n \in \mathbb{Z}$ (will refer to as <u>time</u>)
  - Dependent variable is a real or complex number: $x[n] \in \mathbb{R}$ or $\mathbb{C}$
- Plot signals correctly!



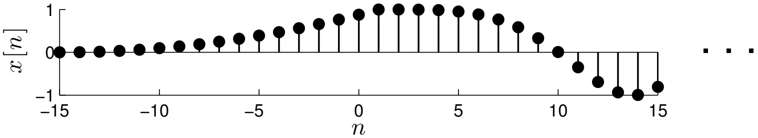
Signal Properties

Signal Properties

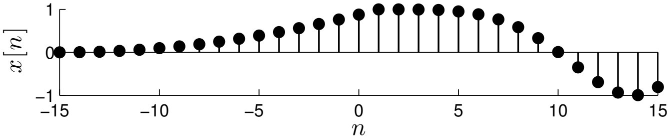
- Real signals
- Complex signals
- Infinite/finite-length signals
- Periodic signals
- Causal signals
- Even/odd signals
- Digital signals

Finite/Infinite-Length Signals

- An **infinite-length** discrete-time signal $x[n]$ is defined for all $n \in \mathbb{Z}$, i.e., $-\infty < n < \infty$



- A **finite-length** discrete-time signal $x[n]$ is defined only for a finite range of $N_1 \leq n \leq N_2$



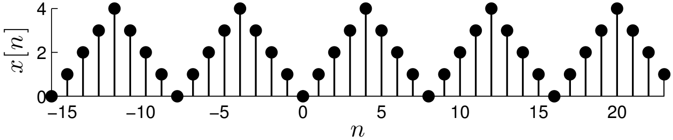
- Important: a finite-length signal is undefined for $n < N_1$ and $n > N_2$

Periodic Signals

DEFINITION

A discrete-time signal is **periodic** if it repeats with period $N \in \mathbb{Z}$:

$$x[n + mN] = x[n] \quad \forall\, m \in \mathbb{Z}$$



Notes:

- The period N must be an integer
- A periodic signal is infinite in length

DEFINITION

A discrete-time signal is **aperiodic** if it is not periodic

Converting between Finite and Infinite-Length Signals

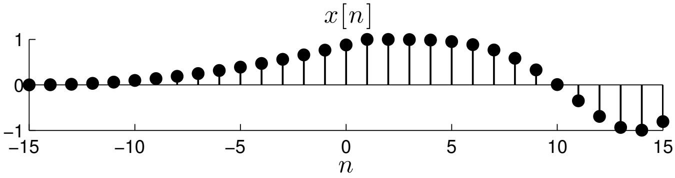
- Convert an infinite-length signal into a finite-length signal by **windowing**

- Convert a finite-length signal into an infinite-length signal by either
  - (infinite) **zero padding**, or
  - **periodization**

Windowing

- Converts a longer signal into a shorter one

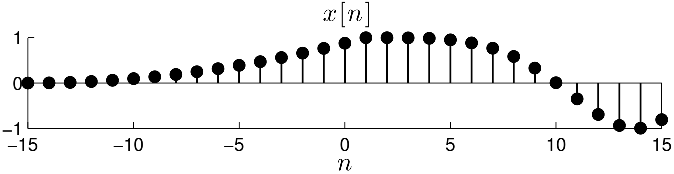
$$y[n] = \begin{cases} x[n] & N_1 \le n \le N_2 \\ 0 & \text{otherwise} \end{cases}$$



Zero Padding

- Converts a shorter signal into a longer one
- Say $x[n]$ is defined for $N_1 \le n \le N_2$
- Given $N_0 \le N_1 \le N_2 \le N_3$

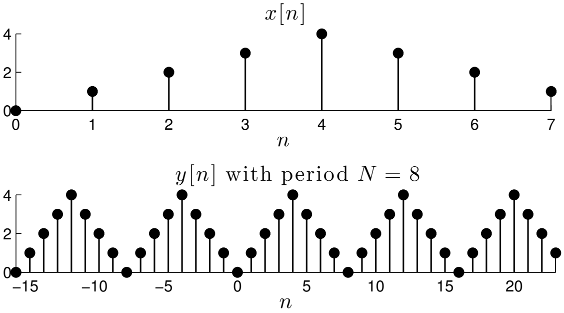
$$y[n] = \begin{cases} 0 & N_0 \le n < N_1 \\ x[n] & N_1 \le n \le N_2 \\ 0 & N_2 \le n \le N_3 \end{cases}$$



Periodization

- Converts a finite-length signal into an infinite-length, periodic signal
- Given finite-length $x[n]$, replicate $x[n]$ periodically with period N

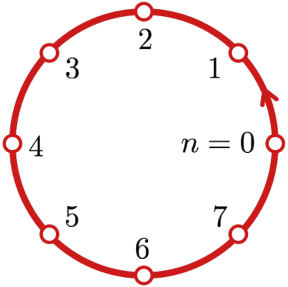
$$\begin{aligned} y[n] &= \sum_{m=-\infty}^{\infty} x[n-mN], \quad n \in \mathbb{Z} \\ &= \cdots + x[n+2N] + x[n+N] + x[n] + x[n-N] + x[n-2N] + \cdots \end{aligned}$$



Useful Aside – Modular Arithmetic

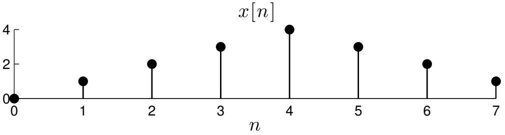
- Modular arithmetic with modulus N (mod-N) takes place on a **clock** with N "hours"
  - Ex: $(12)_8$ ("twelve mod eight")

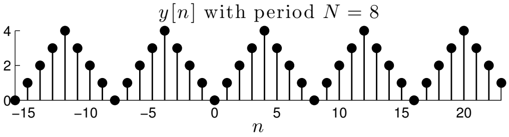
- Modulo arithmetic is inherently periodic
  - Ex: $\ldots\ (-12)_8 \ = \ (-4)_8 \ = \ (4)_8 \ = \ (12)_8 \ = \ (20)_8 \ \ldots$



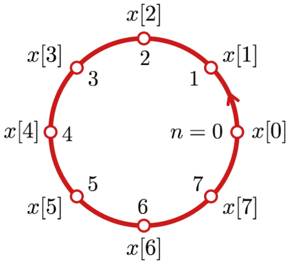
Periodization via Modular Arithmetic

- Consider a length-N signal $x[n]$ defined for $0 \leq n \leq N-1$
- A convenient way to express periodization with period N is $\quad y[n] = x[(n)_N], \quad n \in \mathbb{Z}$

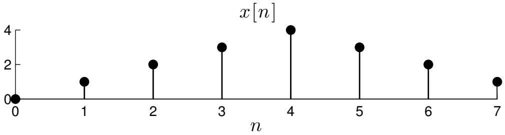


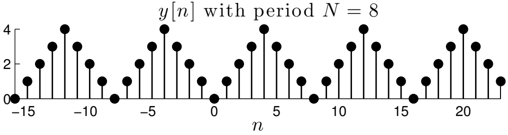


- Important interpretation
  - Infinite-length signals live on the (infinite) number line
  - Periodic signals live on a circle – a clock with N "hours"



Finite-Length and Periodic Signals are Equivalent





- All of the information in a periodic signal is contained in one period (of finite length)

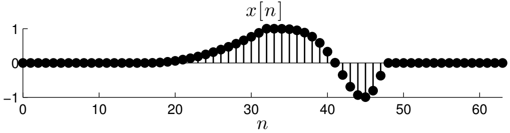
- Any finite-length signal can be periodized

- Conclusion: We can and will think of finite-length signals and periodic signals interchangeably

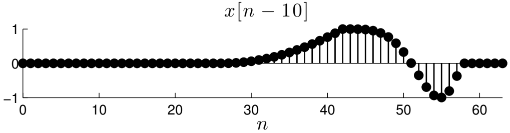
- We can choose the most convenient viewpoint for solving any given problem
  - Application: Shifting finite length signals

Shifting Infinite-Length Signals

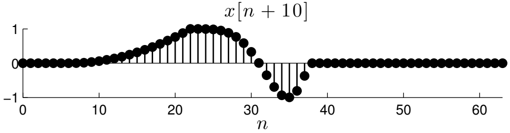
- Given an infinite-length signal $x[n]$, we can **shift** it back and forth in time via $x[n-m]$, $m \in \mathbb{Z}$



- When $m > 0$, $x[n-m]$ shifts to the **right** (forward in time, delay)

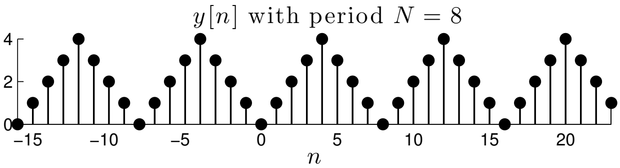


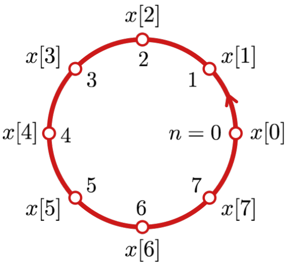
- When $m < 0$, $x[n-m]$ shifts to the **left** (back in time, advance)



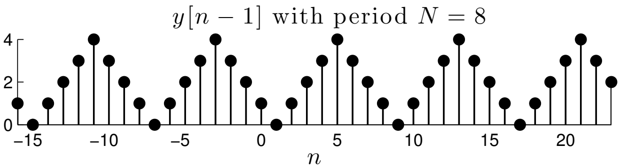
Shifting Periodic Signals

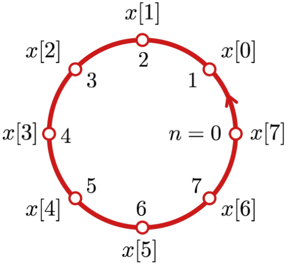
- Periodic signals can also be shifted; consider $y[n] = x[(n)_N]$





- Shift one sample into the future: $y[n-1] = x[(n-1)_N]$





Shifting Finite-Length Signals

- Consider finite-length signals x and v defined for $0 \leq n \leq N-1$ and suppose "$v[n] = x[n-1]$"

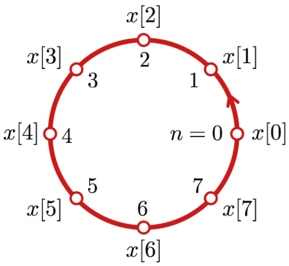
$$\begin{aligned}
v[0] &= ?? \\
v[1] &= x[0] \\
v[2] &= x[1] \\
v[3] &= x[2] \\
&\vdots \\
v[N-1] &= x[N-2] \\
?? &= x[N-1]
\end{aligned}$$

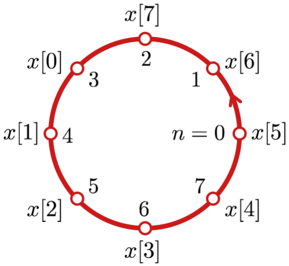
- What to put in $v[0]$? What to do with $x[N-1]$? We don't want to invent/lose information
- Elegant solution: Assume x and v are both periodic with period N; then $v[n] = x[(n-1)_N]$
- This is called a **periodic** or **circular shift** (see `circshift` and `mod` in Matlab)

Circular Shift Example

- Elegant formula for circular shift of $x[n]$ by m time steps: $x[(n-m)_N]$

- Ex: x and v defined for $0 \le n \le 7$, that is, $N = 8$. Find $v[n] = x[(n-3)_8]$



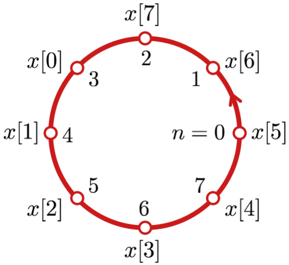


Circular Shift Example

- Elegant formula for circular shift of $x[n]$ by m time steps: $x[(n-m)_N]$

- Ex: x and v defined for $0 \leq n \leq 7$, that is, $N = 8$. Find $v[n] = x[(n-m)_N]$

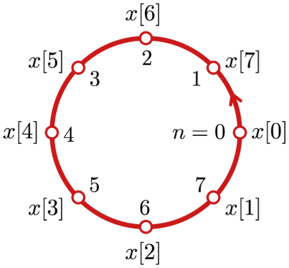
$$\begin{aligned}
v[0] &= x[5] \\
v[1] &= x[6] \\
v[2] &= x[7] \\
v[3] &= x[0] \\
v[4] &= x[1] \\
v[5] &= x[2] \\
v[6] &= x[3] \\
v[7] &= x[4]
\end{aligned}$$



Circular Time Reversal

- For infinite length signals, the transformation of reversing the time axis $x[-n]$ is obvious
- Not so obvious for periodic/finite-length signals
- Elegant formula for reversing the time axis of a periodic/finite-length signal: $x[(-n)_N]$
- Ex: x and v defined for $0 \leq n \leq 7$, that is, $N = 8$. Find $v[n] = x[(-n)_N]$

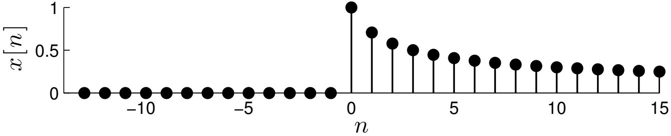
$$\begin{aligned}
v[0] &= x[0] \\
v[1] &= x[7] \\
v[2] &= x[6] \\
v[3] &= x[5] \\
v[4] &= x[4] \\
v[5] &= x[3] \\
v[6] &= x[2] \\
v[7] &= x[1]
\end{aligned}$$



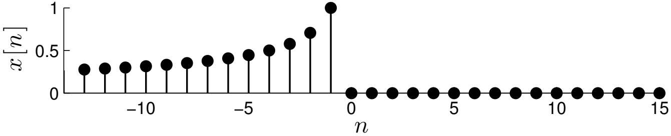
Causal Signals

DEFINITION

A signal $x[n]$ is **causal** if $x[n] = 0$ for all $n < 0$.



- A signal $x[n]$ is **anti-causal** if $x[n] = 0$ for all $n \geq 0$

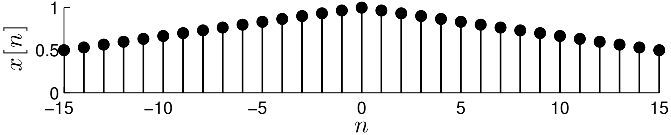


- A signal $x[n]$ is **acausal** if it is not causal

Even Signals

DEFINITION

A real signal $x[n]$ is **even** if $x[-n] = x[n]$

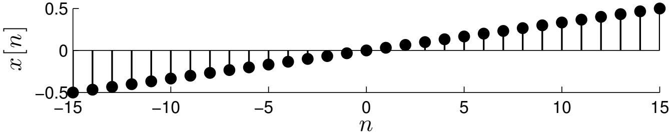


- Even signals are symmetrical around the point $n = 0$

Odd Signals

DEFINITION

A real signal $x[n]$ is **odd** if $x[-n] = -x[n]$



- Even signals are anti-symmetrical around the point $n = 0$

Even+Odd Signal Decomposition

- **Useful fact:** Every signal $x[n]$ can be decomposed into the sum of its even part + its odd part

- Even part: $e[n] = \frac{1}{2}\,(x[n] + x[-n])$ (easy to verify that $e[n]$ is even)

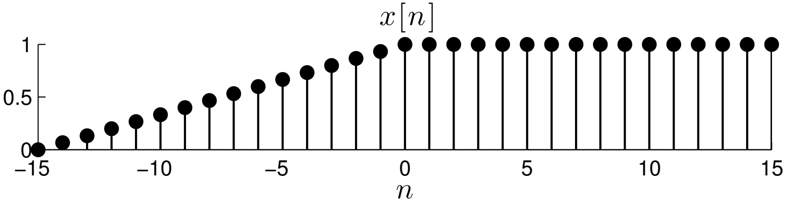
- Odd part: $o[n] = \frac{1}{2}\,(x[n] - x[-n])$ (easy to verify that $o[n]$ is odd)

- **Decomposition** $x[n] = e[n] + o[n]$

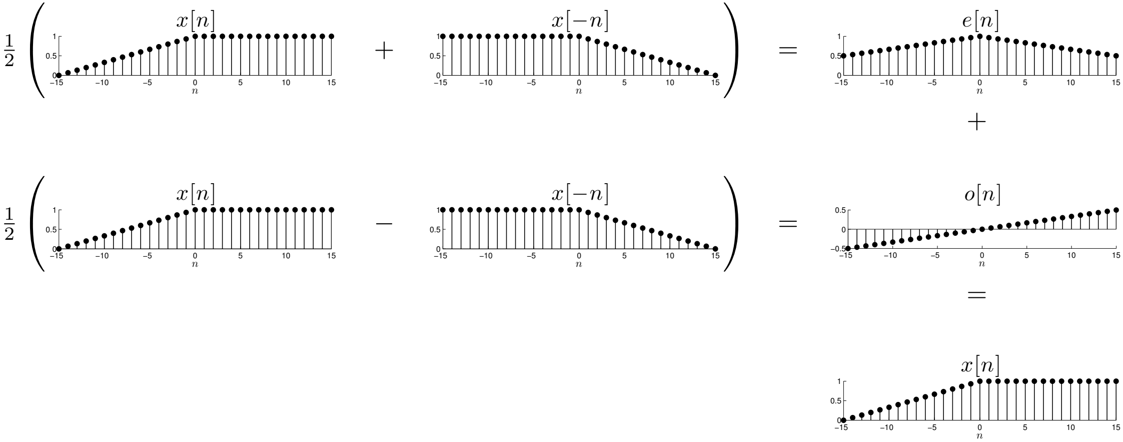
- Verify the decomposition:

$$\begin{aligned} e[n] + o[n] &= \frac{1}{2}(x[n] + x[-n]) + \frac{1}{2}(x[n] - x[-n]) \\ &= \frac{1}{2}(x[n] + x[-n] + x[n] - x[-n]) \\ &= \frac{1}{2}(2\,x[n]) = x[n] \quad \checkmark \end{aligned}$$

Even+Odd Signal Decomposition in Pictures

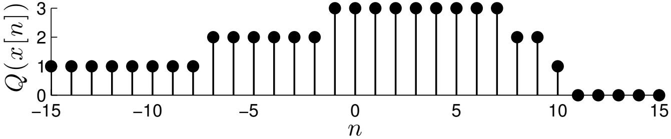


Even+Odd Signal Decomposition in Pictures



Digital Signals

- **Digital signals** are a special sub-class of discrete-time signals
  - Independent variable is still an integer: $n \in \mathbb{Z}$
  - Dependent variable is from a **finite set of integers**: $x[n] \in \{0, 1, \ldots, D-1\}$
  - Typically, choose $D = 2^q$ and represent each possible level of $x[n]$ as a digital code with q bits
  - Ex: Digital signal with $q = 2$ bits $\Rightarrow D = 2^2 = 4$ levels

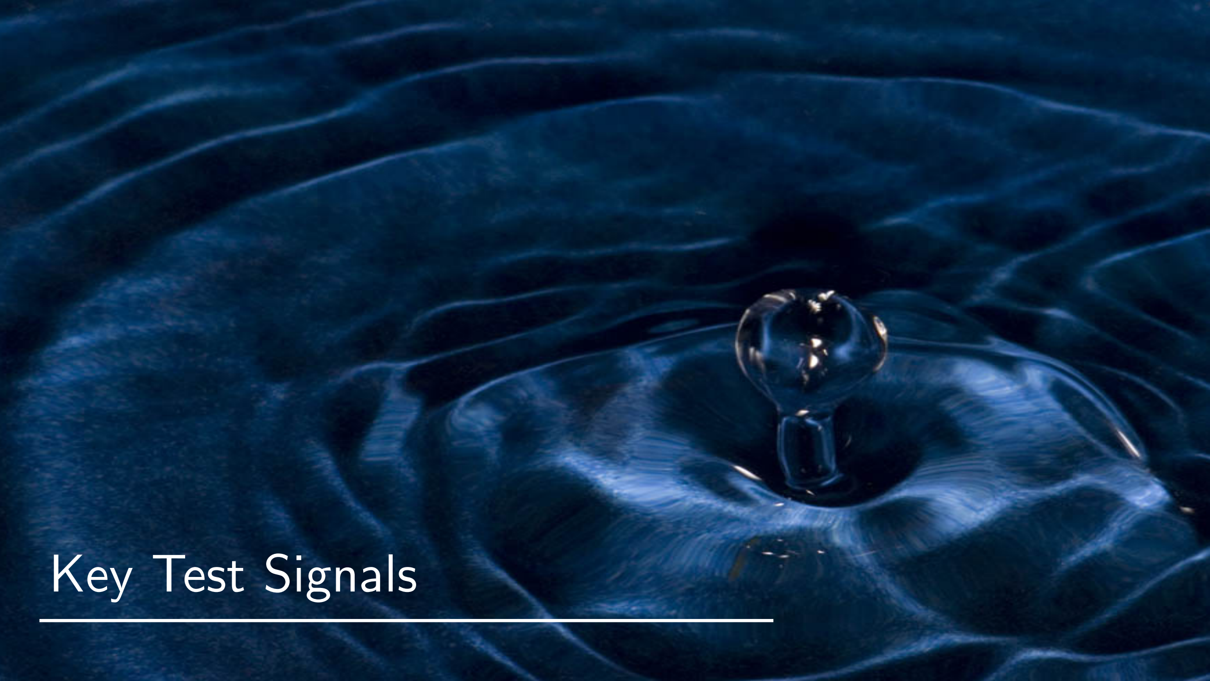


  - Ex: Compact discs use $q = 16$ bits $\Rightarrow D = 2^{16} = 65536$ levels

Summary

- Signals can be classified many different ways (real/complex, infinite/finite-length, periodic/aperiodic, causal/acausal, even/odd, . . .)

- Finite-length signals are equivalent to periodic signals, modulo arithmetic useful



Key Test Signals

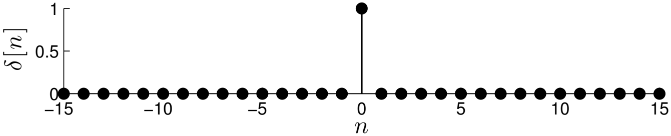
A Toolbox of Test Signals

- Delta function
- Unit step
- Unit pulse (boxcar)
- Real exponential
- Next lecture
  - Sinusoids
  - (Complex) sinusoid
  - Complex exponential
- **Note:** We will introduce the test signals as infinite-length signals, but each has a finite-length equivalent

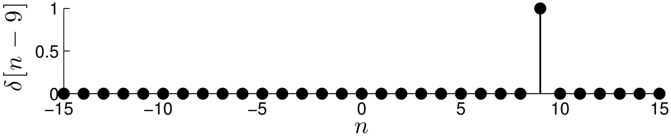
Delta Function

DEFINITION

The **delta function** (aka unit impulse) $\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$



- The shifted delta function $\delta[n-m]$ peaks up at $n = m$; here $m = 9$

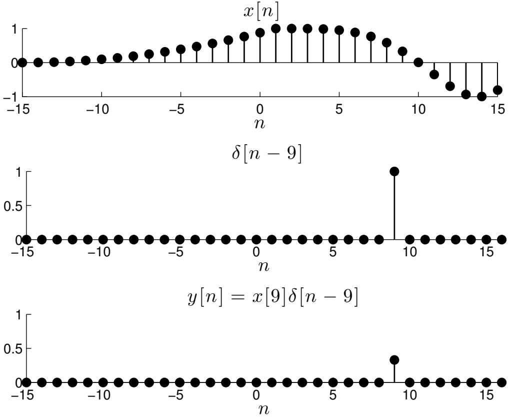


Delta Functions Sample

- Multiplying a signal by a shifted delta function picks out one sample of the signal and sets all other samples to zero

$$y[n] \;=\; x[n]\,\delta[n-m] \;=\; x[m]\,\delta[n-m]$$

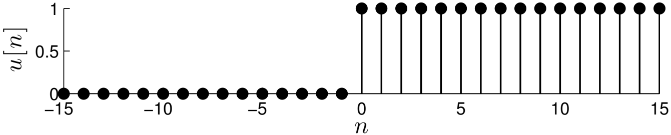
- Important: m is a fixed constant, and so $x[m]$ is a constant (and not a signal)



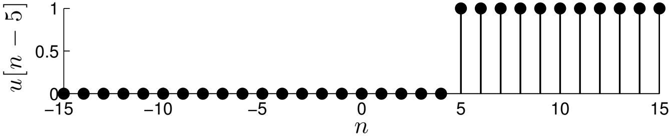
Unit Step

DEFINITION

The **unit step** $u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$



- The shifted unit step $u[n-m]$ jumps from 0 to 1 at $n = m$; here $m = 5$

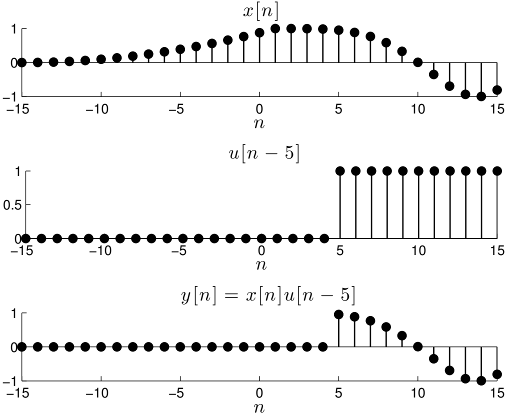


Unit Step Selects Part of a Signal

- Multiplying a signal by a shifted unit step function zeros out its entries for $n < m$

$$y[n] \;=\; x[n]\,u[n-m]$$

(Note: For $m =$, this makes $y[n]$ causal)

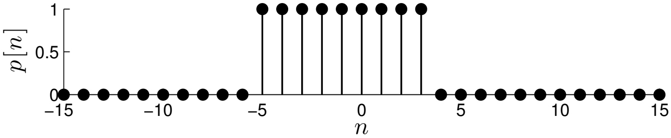


Unit Pulse (Boxcar)

DEFINITION

The **unit pulse** (boxcar) $p[n] = \begin{cases} 0 & n < N_1 \\ 1 & N_1 \le n \le N_2 \\ 0 & n > N_2 \end{cases}$

- Ex: $p[n]$ for $N_1 = -5$ and $N_2 = 3$



- One of many different formulas for the unit pulse

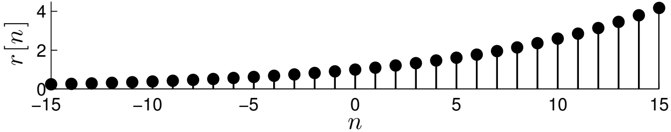
$$p[n] = u[n - N_1] - u[n - (N_2 + 1)]$$

Real Exponential

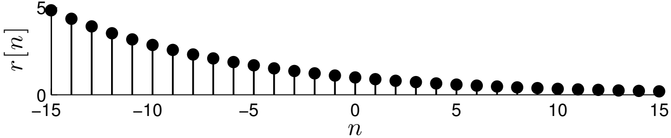
DEFINITION

The **real exponential** $r[n] = a^n$, $a \in \mathbb{R}$, $a \geq 0$

- For $a > 1$, $r[n]$ shrinks to the left and grows to the right; here $a = 1.1$



- For $0 < a < 1$, $r[n]$ grows to the left and shrinks to the right; here $a = 0.9$



Summary

- We will use our test signals a lot, especially the delta function and unit step

Sinusoids

Table of Contents

- Lecture in four parts:
  - Part 1: Real and Complex Sinusoids
  - Part 2: Sinusoids are Weird: Aliasing
  - Part 3: Sinusoids are Weird: Periodicity
  - Part 4: Complex Exponentials

Sinusoids, Part 1
Real and Complex Sinusoids

A Toolbox of Test Signals, Cont'd

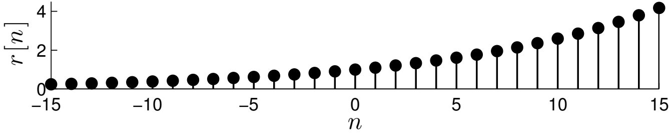
- Sinusoids appear in myriad disciplines, in particular signal processing
- They are the basis (literally) of Fourier analysis (DFT, DTFT)
- We will introduce
  - Real-valued sinusoids
  - (Complex) sinusoid
  - Complex exponential

Recall: Real Exponential

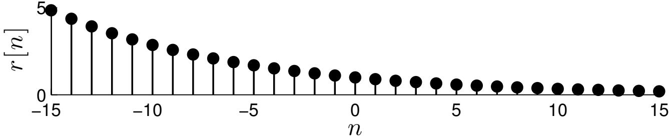
DEFINITION

The **real exponential** $r[n] = a^n$, $a \in \mathbb{R}$, $a \geq 0$

- For $a > 1$, $r[n]$ shrinks to the left and grows to the right; here $a = 1.1$



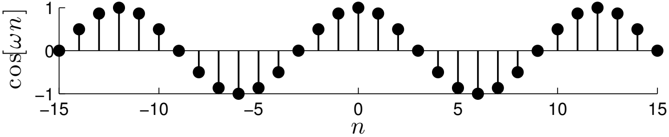
- For $0 < a < 1$, $r[n]$ grows to the left and shrinks to the right; here $a = 0.9$



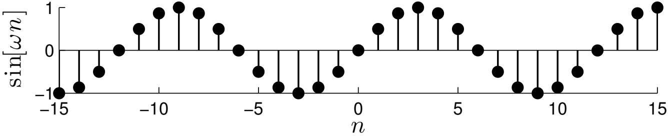
Sinusoids

- There are two natural real-valued sinusoids: $\cos(\omega n + \phi)$ and $\sin(\omega n + \phi)$
- **Frequency:** ω (units: radians/sample)
- **Phase:** ϕ (units: radians)

- $\cos(\omega n)$ (even)

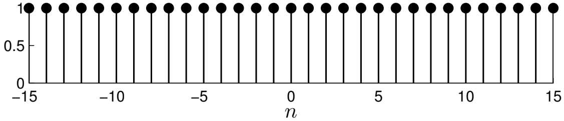


- $\sin(\omega n)$ (odd)

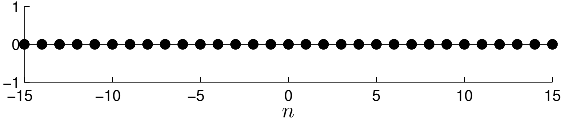


Sinusoid Examples

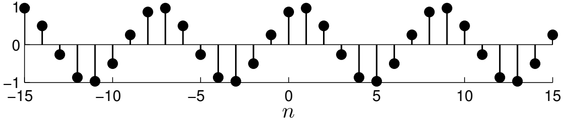
- $\cos(0n)$



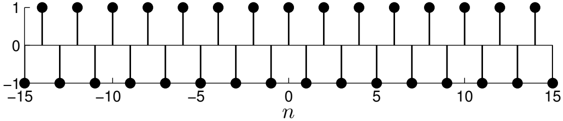
- $\sin(0n)$



- $\sin(\frac{\pi}{4}n + \frac{2\pi}{6})$



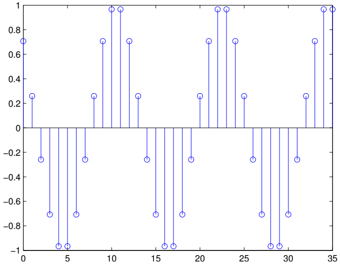
- $\cos(\pi n)$



Get Comfortable with Sinusoids!

- It's easy to play around in Matlab to get comfortable with the properties of sinusoids

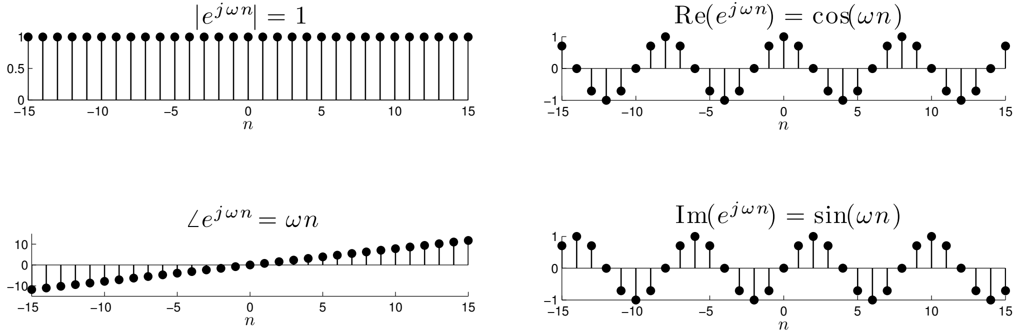
```
N=36;
n=0:N-1;
omega=pi/6;
phi=pi/4;
x=cos(omega*n+phi);
stem(n,x)
```



Complex Sinusoid

- The complex-valued sinusoid combines both the $\cos$ and $\sin$ terms (via Euler's identity)

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



$|e^{j\omega n}| = 1$

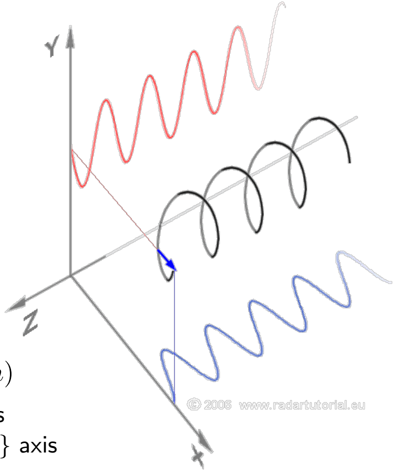
$\mathrm{Re}(e^{j\omega n}) = \cos(\omega n)$

$\angle e^{j\omega n} = \omega n$

$\mathrm{Im}(e^{j\omega n}) = \sin(\omega n)$

A Complex Sinusoid is a Helix

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



- A complex sinusoid is a **helix** in 3D space $(\mathrm{Re}\{\}, \mathrm{Im}\{\}, n)$
  - Real part ($\cos$ term) is the projection onto the $\mathrm{Re}\{\}$ axis
  - Imaginary part ($\sin$ term) is the projection onto the $\mathrm{Im}\{\}$ axis
- Frequency ω determines rotation speed and direction of helix
  - $\omega > 0 \Rightarrow$ anticlockwise rotation
  - $\omega < 0 \Rightarrow$ clockwise rotation

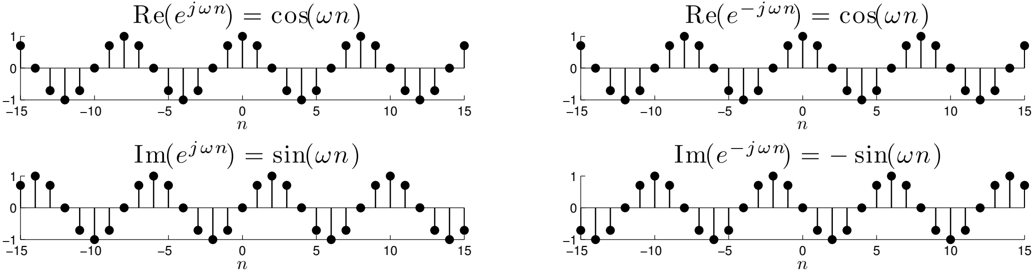
Negative Frequency

- Negative frequency is nothing to be afraid of! Consider a sinusoid with a negative frequency $-\omega$

$$e^{j(-\omega)n} \;=\; e^{-j\omega n} \;=\; \cos(-\omega n) + j\sin(-\omega n) \;=\; \cos(\omega n) - j\sin(\omega n)$$

- Also note: $e^{j(-\omega)n} \;=\; e^{-j\omega n} \;=\; \left(e^{j\omega n}\right)^*$

- Bottom line: negating the frequency is equivalent to complex conjugating a complex sinusoid, which flips the sign of the imaginary, $\sin$ term



$\mathrm{Re}(e^{j\omega n}) = \cos(\omega n)$

$\mathrm{Re}(e^{-j\omega n}) = \cos(\omega n)$

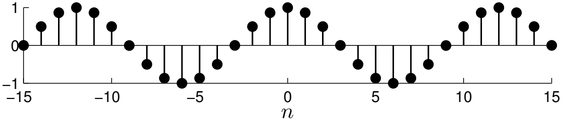
$\mathrm{Im}(e^{j\omega n}) = \sin(\omega n)$

$\mathrm{Im}(e^{-j\omega n}) = -\sin(\omega n)$

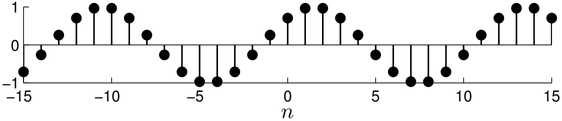
Phase of a Sinusoid $e^{j(\omega n+\phi)}$

- ϕ is a (frequency independent) shift that is referenced to one period of oscillation

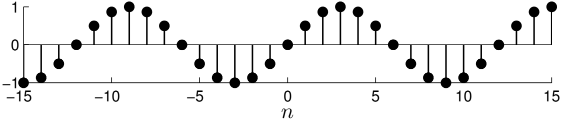
- $\cos\left(\frac{\pi}{6}n - 0\right)$



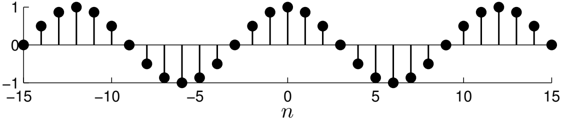
- $\cos\left(\frac{\pi}{6}n - \frac{\pi}{4}\right)$

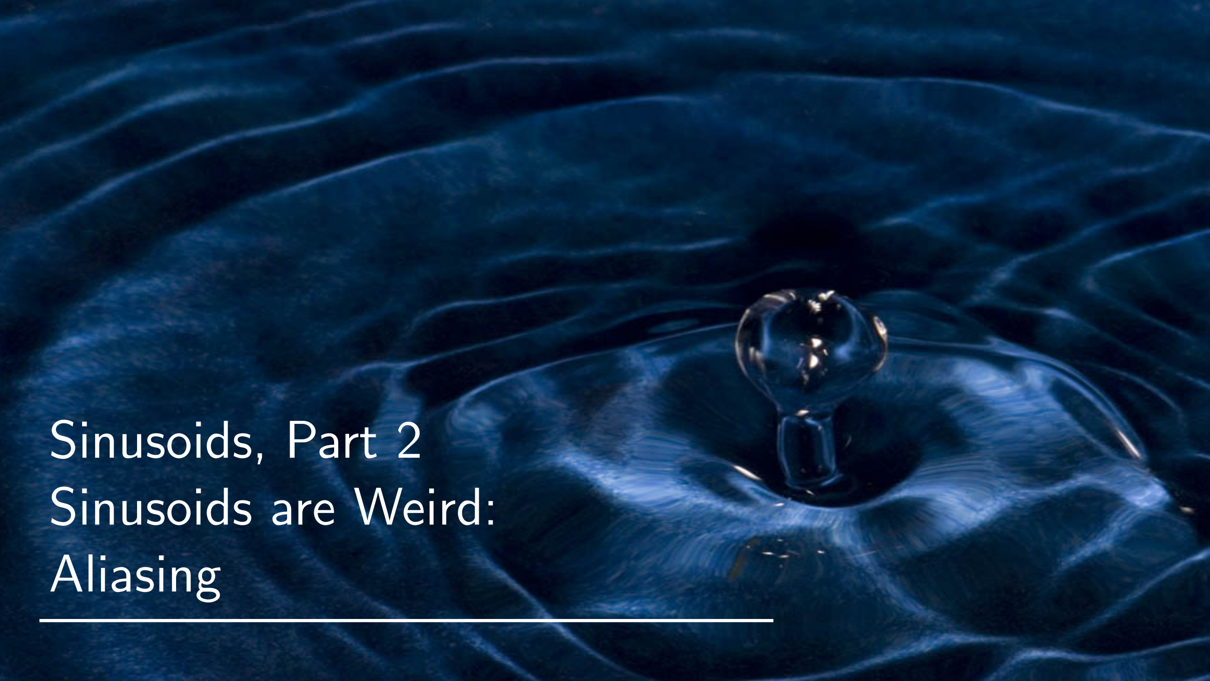


- $\cos\left(\frac{\pi}{6}n - \frac{\pi}{2}\right) = \sin\left(\frac{\pi}{6}n\right)$



- $\cos\left(\frac{\pi}{6}n - 2\pi\right) = \cos\left(\frac{\pi}{6}n\right)$





Sinusoids, Part 2
Sinusoids are Weird: Aliasing

Discrete-Time Sinusoids are Weird!

- Discrete-time sinusoids $e^{j(\omega n+\phi)}$ have two counterintuitive properties

- Both involve the frequency ω

Discrete-Time Sinusoids are Weird!

- Weird property #1: ALIASING

Aliasing of Sinusoids

- Consider two sinusoids with two different frequencies
  - $\omega \quad \Rightarrow \quad x_1[n] = e^{j(\omega n+\phi)}$
  - $\omega + 2\pi \quad \Rightarrow \quad x_2[n] = e^{j((\omega+2\pi)n+\phi)}$

- But note that

$$x_2[n] = e^{j(\omega+2\pi)n+\phi)} = e^{j(\omega n+\phi)+j2\pi n} = e^{j(\omega n+\phi)}\ e^{j2\pi n} = e^{j(\omega n+\phi)} = x_1[n]$$

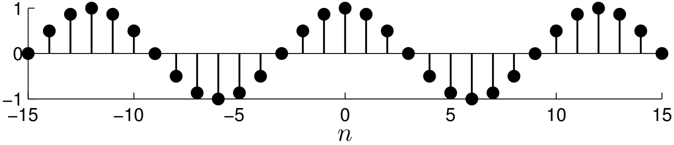
- The signals x_1 and x_2 have different frequencies but are **identical**!

- We say that x_1 and x_2 are aliases; this phenomenon is called **aliasing**

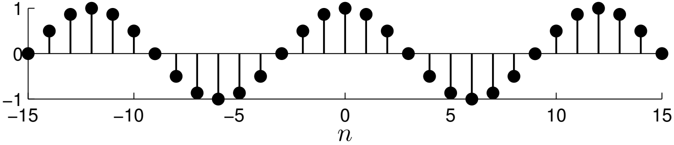
- Note: Any integer multiple of 2π will do; try with $x_3[n] = e^{j((\omega+2\pi m)n+\phi)}$, $m \in \mathbb{Z}$

Aliasing of Sinusoids – Example

- $x_1[n] = \cos\left(\frac{\pi}{6}n\right)$



- $x_2[n] = \cos\left(\frac{13\pi}{6}n\right) = \cos\left((\frac{\pi}{6} + 2\pi)n\right)$



Alias-Free Frequencies

- Since

  $$x_3[n] = e^{j(\omega+2\pi m)n+\phi)} = e^{j(\omega n+\phi)} = x_1[n] \quad \forall m \in \mathbb{Z}$$

  the only frequencies that lead to unique (distinct) sinusoids lie in an interval of length 2π

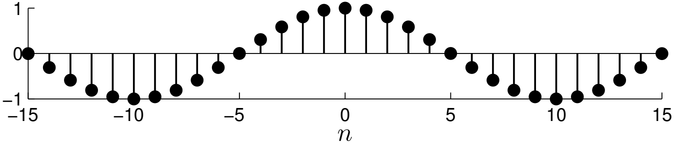
- Convenient to interpret the frequency ω as an **angle**
  (then aliasing is handled automatically; more on this later)

- Two intervals are typically used in the signal processing literature (and in this course)
  - $0 \leq \omega < 2\pi$
  - $-\pi < \omega \leq \pi$

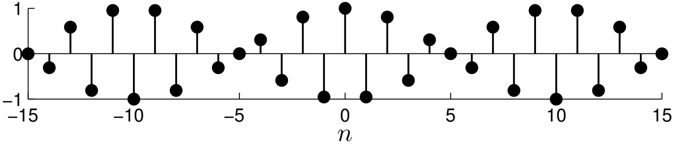
Low and High Frequencies

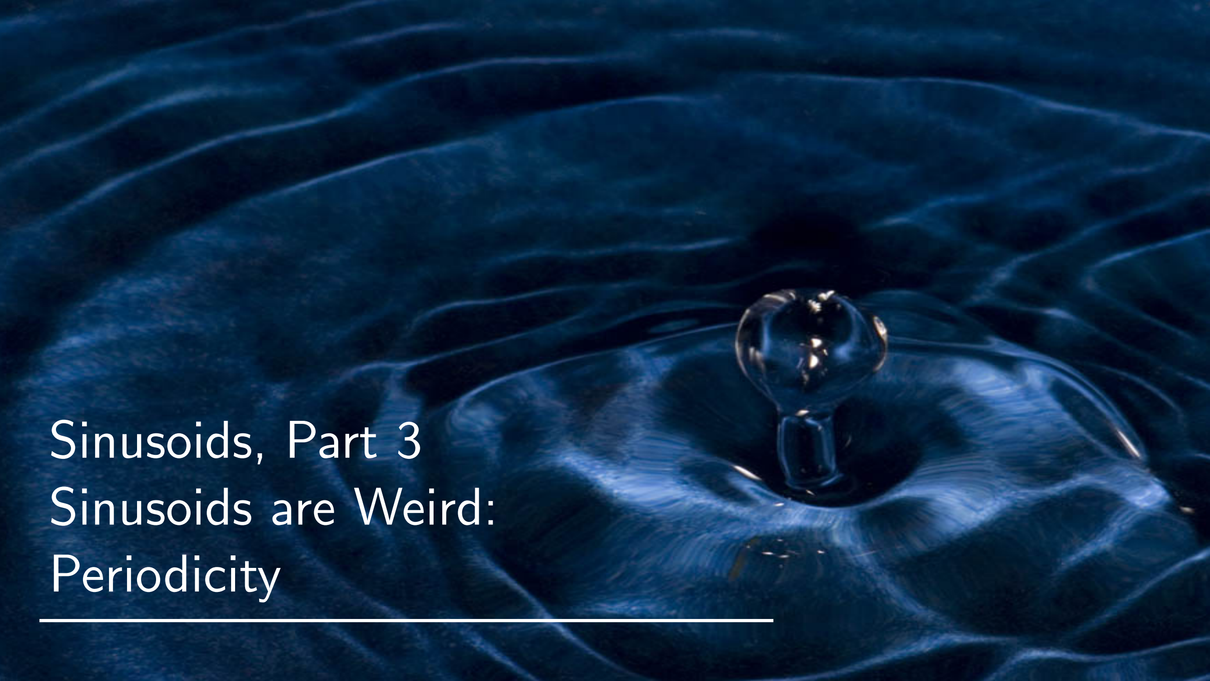
$$e^{j(\omega n+\phi)}$$

- **Low frequencies:** ω close to 0 or 2π rad
  Ex: $\cos\left(\frac{\pi}{10}n\right)$



- **High frequencies:** ω close to π or $-\pi$ rad
  Ex: $\cos\left(\frac{9\pi}{10}n\right)$





Sinusoids, Part 3
Sinusoids are Weird: Periodicity

Discrete-Time Sinusoids are Weird!

- Weird property #2: PERIODICITY?

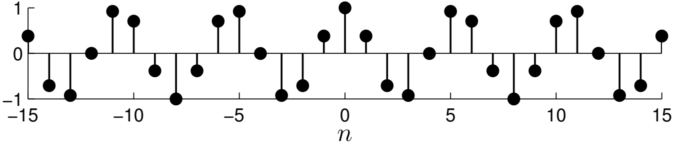
Periodicity of Sinusoids

- Consider $x_1[n] = e^{j(\omega n+\phi)}$ with frequency $\omega = \frac{2\pi k}{N}$, $\quad k, N \in \mathbb{Z}$ (harmonic frequency)

- It is easy to show that x_1 is periodic with period N, since

$$x_1[n+N] = e^{j(\omega(n+N)+\phi)} = e^{j(\omega n+\omega N+\phi)} = e^{j(\omega n+\phi)}\, e^{j(\omega N)} = e^{j(\omega n+\phi)}\, e^{j(\frac{2\pi k}{N} N)} = x_1[n] \checkmark$$

- Ex: $x_1[n] = \cos(\frac{2\pi 3}{16} n)$, $\quad N = 16$



- Note: x_1 is periodic with the (smaller) period of $\frac{N}{k}$ when $\frac{N}{k}$ is an integer

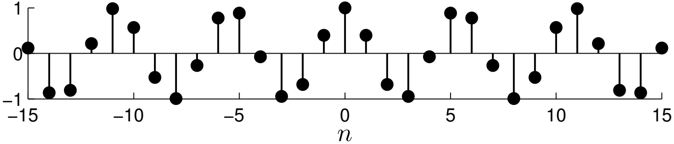
Aperiodicity of Sinusoids

- Consider $x_2[n] = e^{j(\omega n+\phi)}$ with frequency $\omega \neq \frac{2\pi k}{N}, \quad k, N \in \mathbb{Z}$ (not harmonic frequency)

- Is x_2 periodic?

$$x_2[n+N] = e^{j(\omega(n+N)+\phi)} = e^{j(\omega n+\omega N+\phi)} = e^{j(\omega n+\phi)}\, e^{j(\omega N)} \neq x_1[n] \quad \text{NO!}$$

- Ex: $x_2[n] = \cos(1.16\, n)$



- If its frequency ω is not harmonic, then a sinusoid oscillates but is not periodic!

Harmonic Sinusoids

$$e^{j(\omega n+\phi)}$$

- Semi-amazing fact: The **only** periodic discrete-time sinusoids are those with harmonic frequencies

$$\omega = \frac{2\pi k}{N}, \quad k, N \in \mathbb{Z}$$

- Which means that
  - **Most** discrete-time sinusoids are **not** periodic!
  - The harmonic sinusoids are somehow magical (they play a starring role later in the DFT)

Sinusoids, Part 4
Complex Exponentials

Complex Exponential

- Complex sinusoid $e^{j(\omega n+\phi)}$ is of the form $e^{\text{Purely Imaginary Numbers}}$

- Generalize to $e^{\text{General Complex Numbers}}$

- Consider the general complex number $z = |z|\, e^{j\omega}$, $z \in \mathbb{C}$
  - $|z| =$ magnitude of z
  - $\omega = \angle(z)$, phase angle of z
  - Can visualize $z \in \mathbb{C}$ as a **point** in the **complex plane**

- Now we have

$$z^n = (|z|e^{j\omega})^n = |z|^n (e^{j\omega})^n = |z|^n e^{j\omega n}$$

  - $|z|^n$ is a **real exponential** $(a^n$ with $a = |z|)$
  - $e^{j\omega n}$ is a **complex sinusoid**

Complex Exponential is a Spiral

$$z^n \;=\; \left(|z|\, e^{j\omega n}\right)^n \;=\; |z|^n\, e^{j\omega n}$$

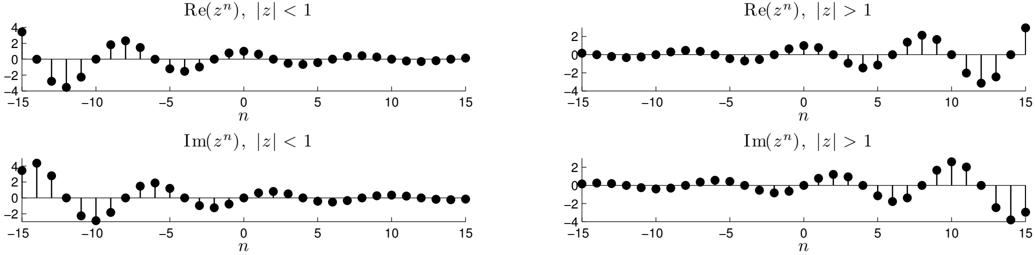
- $|z|^n$ is a **real exponential** envelope $(a^n$ with $a = |z|)$
- $e^{j\omega n}$ is a **complex sinusoid**
- z^n is a helix with expanding radius (spiral)

Complex Exponential is a Spiral

$$z^n \;=\; \left(|z|\,e^{j\omega n}\right)^n \;=\; |z|^n\,e^{j\omega n}$$

- $|z|^n$ is a **real exponential** envelope $(a^n$ with $a = |z|)$
- $e^{j\omega n}$ is a **complex sinusoid**

$|z| < 1$ $|z| > 1$



Summary

- We will use our test signals a lot, especially the sinusoids
- Discrete-time sinusoids **alias**; as a result, the only unique frequencies lie in a range of length 2π
- Discrete-time sinusoids **oscillate** but are only **periodic** when the frequency is harmonic

Signals are Vectors

Table of Contents

- Lecture in three parts:
  - Part 1: Vector Spaces
  - Part 2: Linear Combination + Matlab Demo
  - Part 3: Strength of a Vector

Signals are Vectors, Part 1: Vector Space

Signals are Vectors

- Signals are mathematical objects
- Here we will develop tools to analyze the **geometry** of sets of signals
- The tools come from **linear algebra**
- By interpreting signals as vectors in a vector space, we will be able to speak about the length of a signal (its "strength," more below), angles between signals (their similarity), and more
- We will also be able to use matrices to better understand how signal processing systems work
- Caveat: This is not a course on linear algebra!

Vector Space

DEFINITION

A linear **vector space** V is a collection of vectors such that if $x, y \in V$ and α is a scalar then

$$\alpha\, x \in V \qquad \text{and} \qquad x + y \in V$$

- In words:
  - A rescaled vector stays in the vector space
  - The sum of two vectors stays in the vector space

- We will be interested in scalars (basically, numbers) α that either live in $\mathbb{R}$ or $\mathbb{C}$

- Classical vector spaces that you know and love
  - $\mathbb{R}^N$, the set of all vectors of length N with real-valued entries
  - $\mathbb{C}^N$, the set of all vectors of length N with complex-valued entries
  - Special case that we will use all the time to draw pictures and build intuition: $\mathbb{R}^2$

The Vector Space $\mathbb{R}^2$ (1)

- Vectors in $\mathbb{R}^2$: $x = \begin{bmatrix} x[0] \\ x[1] \end{bmatrix}, \;\; y = \begin{bmatrix} y[0] \\ y[1] \end{bmatrix}, \;\; x[0], x[1], y[0], y[1] \in \mathbb{R}$
  - Note: We will enumerate the entries of a vector starting from 0 rather than 1 (this is the convention in signal processing and programming languages like "C", but not in Matlab)
  - Note: We will not use the traditional boldface or underline notation for vectors

- Scalars: $\alpha \in \mathbb{R}$

- Scaling: $\alpha\, x = \alpha \begin{bmatrix} x[0] \\ x[1] \end{bmatrix} = \begin{bmatrix} \alpha\, x[0] \\ \alpha\, x[1] \end{bmatrix}$

The Vector Space $\mathbb{R}^2$ (2)

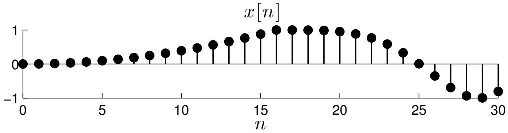
- Vectors in $\mathbb{R}^2$: $x = \begin{bmatrix} x[0] \\ x[1] \end{bmatrix}, \quad y = \begin{bmatrix} y[0] \\ y[1] \end{bmatrix}, \; x[0], x[1], y[0], y[1] \in \mathbb{R}$

- Scalars: $\alpha \in \mathbb{R}$

- Summing: $x + y = \begin{bmatrix} x[0] \\ x[1] \end{bmatrix} + \begin{bmatrix} y[0] \\ y[1] \end{bmatrix} = \begin{bmatrix} x[0] + y[0] \\ x[1] + y[1] \end{bmatrix}$

The Vector Space $\mathbb{R}^N$

- Vectors in $\mathbb{R}^N$: $x = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}, \quad x[n] \in \mathbb{R}$

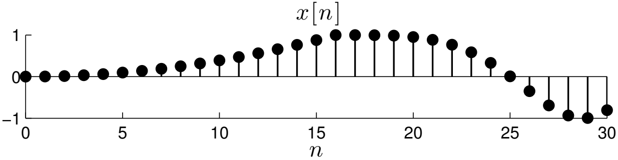


- This is exactly the same as a real-valued discrete time signal; that is, **signals are vectors**
  - Scaling αx amplifies/attenuates a signal by the factor α
  - Summing $x + y$ creates a new signal that mixes x and y

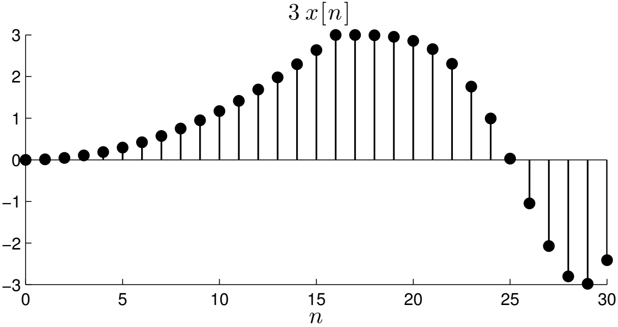
- $\mathbb{R}^N$ is harder to visualize than $\mathbb{R}^2$ and $\mathbb{R}^3$, but the intuition gained from $\mathbb{R}^2$ and $\mathbb{R}^3$ generally holds true with no surprises (at least in this course)

The Vector Space $\mathbb{R}^N$ – Scaling

- Signal $x[n]$

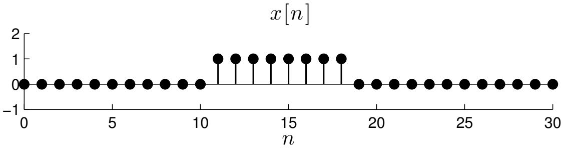


- Scaled signal $3\,x[n]$

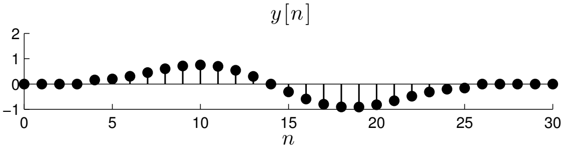


The Vector Space $\mathbb{R}^N$ – Summing

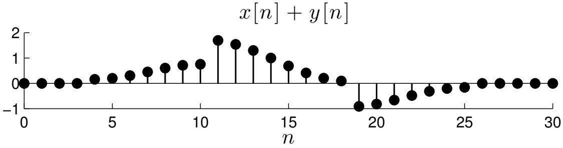
- Signal $x[n]$



- Signal $y[n]$



- Sum $x[n] + y[n]$



The Vector Space $\mathbb{C}^N$ (1)

- $\mathbb{C}^N$ is the same as $\mathbb{R}^N$ with a few minor modifications

- Vectors in $\mathbb{C}^N$: $x = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}, \;\; x[n] \in \mathbb{C}$

- Each entry $x[n]$ is a complex number that can be represented as

$$x[n] = \mathrm{Re}\{x[n]\} + j\,\mathrm{Im}\{x[n]\} = |x[n]|\, e^{j\angle x[n]}$$

- Scalars $\alpha \in \mathbb{C}$

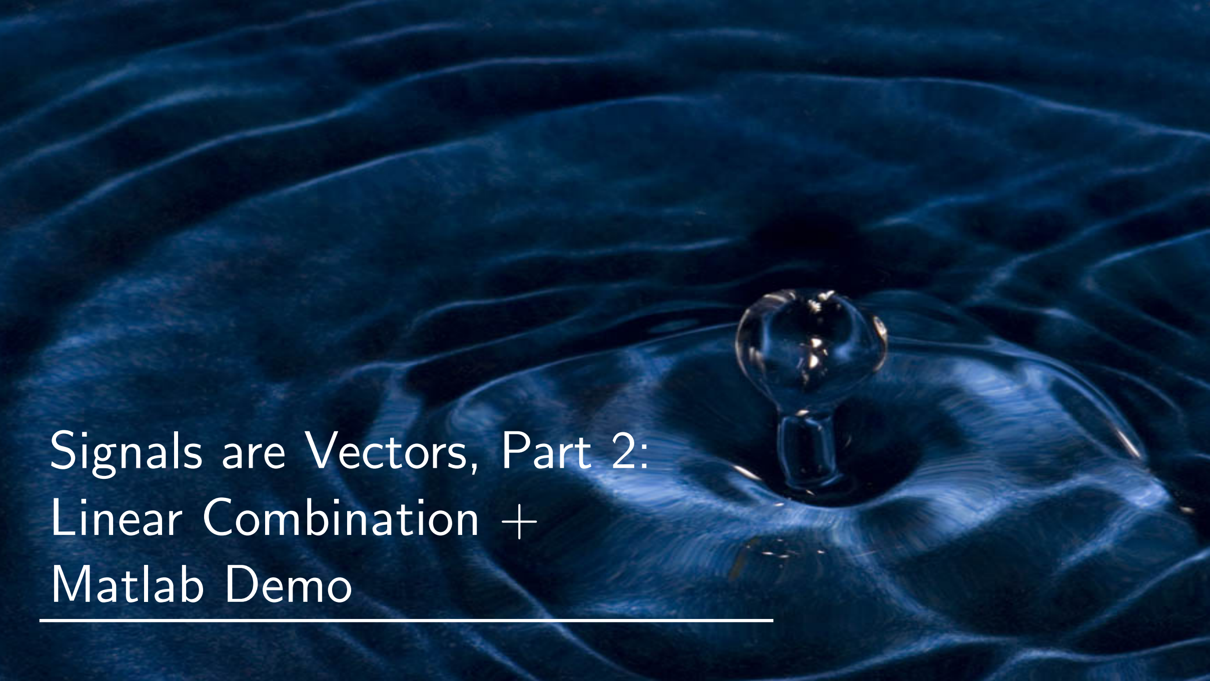
The Vector Space $\mathbb{C}^N$ (2)

- Rectangular form

$$x = \begin{bmatrix} \mathrm{Re}\{x[0]\} + j\,\mathrm{Im}\{x[0]\} \\ \mathrm{Re}\{x[1]\} + j\,\mathrm{Im}\{x[1]\} \\ \vdots \\ \mathrm{Re}\{x[N-1]\} + j\,\mathrm{Im}\{x[N-1]\} \end{bmatrix} = \mathrm{Re}\left\{ \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \right\} + j\,\mathrm{Im}\left\{ \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \right\}$$

- Polar form

$$x = \begin{bmatrix} |x[0]|\, e^{j\angle x[0]} \\ |x[1]|\, e^{j\angle x[1]} \\ \vdots \\ |x[N-1]|\, e^{j\angle x[N-1]} \end{bmatrix}$$



Signals are Vectors, Part 2: Linear Combination + Matlab Demo

Linear Combination

DEFINITION

Given a collection of M vectors $x_0, x_1, \ldots x_{M-1} \in \mathbb{C}^N$ and M scalars $\alpha_0, \alpha_1, \ldots, \alpha_{M-1} \in \mathbb{C}$, the **linear combination** of the vectors is given by

$$y = \alpha_0 x_0 + \alpha_1 x_1 + \cdots + \alpha_{M-1} x_{M-1} = \sum_{m=0}^{M-1} \alpha_m x_m$$

- Clearly the result of the linear combination is a vector $y \in \mathbb{C}^N$

Linear Combination Example

- A recording studio uses a **mixing board** (or desk) to create a linear combination of the signals from the different instruments that make up a song

- Say $x_0 =$ drums, $x_1 =$ bass, $x_2 =$ guitar, $\ldots$, $x_{22} =$ saxophone, $x_{23} =$ singer

- Linear combination (output of mixing board)

$$y = \alpha_0 x_0 + \alpha_1 x_1 + \cdots + \alpha_{M-1} x_{M-1} = \sum_{m=0}^{M-1} \alpha_m x_m$$

- Changing the α_m's results in a different "mix" y that emphasizes/deemphasizes certain instruments

Linear Combination = Matrix Multiplication

- Step 1: Stack the vectors $x_m \in \mathbb{C}^N$ as column vectors into an $N \times M$ matrix

$$X = \big[x_0|x_1|\cdots|x_{M-1}\big]$$

- Step 2: Stack the scalars α_m into an $M \times 1$ column vector

$$a = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{M-1} \end{bmatrix}$$

- Step 3: We can now write a linear combination as the matrix/vector product

$$y \;=\; \alpha_0 x_0 + \alpha_1 x_1 + \cdots + \alpha_{M-1} x_{M-1} \;=\; \sum_{m=0}^{M-1} \alpha_m x_m \;=\; \big[x_0|x_1|\cdots|x_{M-1}\big] \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{M-1} \end{bmatrix} \;=\; Xa$$

Linear Combination = Matrix Multiplication (The Gory Details)

- M vectors in $\mathbb{C}^N$: $x_m = \begin{bmatrix} x_m[0] \\ x_m[1] \\ \vdots \\ x_m[N-1] \end{bmatrix}, \quad m = 0, 1, \ldots, M-1$

- $N \times M$ matrix: $X = \begin{bmatrix} x_0[0] & x_1[0] & \cdots & x_{M-1}[0] \\ x_0[1] & x_1[1] & \cdots & x_{M-1}[1] \\ \vdots & \vdots & & \vdots \\ x_0[N-1] & x_1[N-1] & \cdots & x_{M-1}[N-1] \end{bmatrix}$

- Note: The row-n, column-m element of the matrix $[X]_{n,m} = x_m[n]$

- M scalars α_m, $m = 0, 1, \ldots, M-1$: $a = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{M-1} \end{bmatrix}$

- Linear combination $y = Xa$

Linear Combination = Matrix Multiplication (Summary)

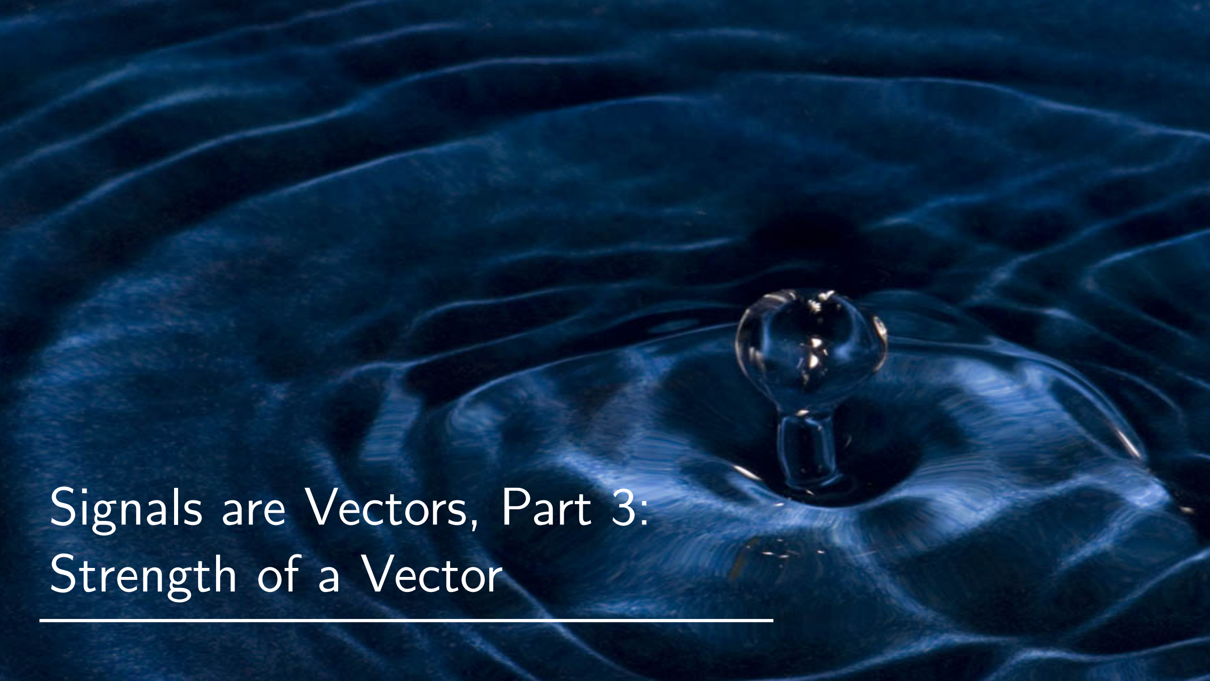
- Linear combination $y = Xa$

- The row-n, column-m element of the $N \times M$ matrix $[X]_{n,m} = x_m[n]$

$$y = \begin{bmatrix} \vdots \\ y[n] \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & x_m[n] & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \alpha_m \\ \vdots \end{bmatrix} = Xa$$

- Sum-based formula for $y[n]$

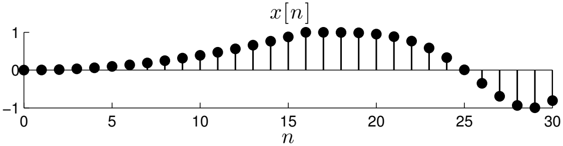
$$y[n] = \sum_{m=0}^{M-1} \alpha_m \, x_m[n]$$



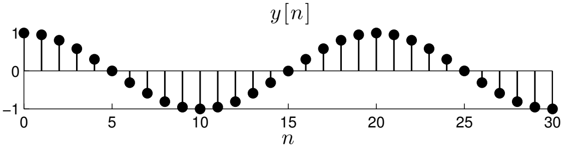
Signals are Vectors, Part 3: Strength of a Vector

Strength of a Vector

- How to quantify the "strength" of a vector?
- How to say that one signal is "stronger" than another?
- Signal x



- Signal y



Strength of a Vector: 2-Norm

DEFINITION

The **Euclidean length**, or 2-**norm**, of a vector $x \in \mathbb{C}^N$ is given by

$$\|x\|_2 = \sqrt{\sum_{n=0}^{N-1} |x[n]|^2}$$

The **energy** of x is given by $(\|x\|_2)^2 = \|x\|_2^2$

- The norm takes as input a vector in $\mathbb{C}^N$ and produces a real number that is ≥ 0
- When it is clear from context, we will suppress the subscript "2" in $\|x\|_2$ and just write $\|x\|$

2-Norm Example

- Ex: $x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

- ℓ_2 norm

$$\|x\|_2 \; = \; \sqrt{\sum_{n=0}^{N-1} |x[n]|^2} \; = \; \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$$

Strength of a Vector: p-Norm

- The Euclidean length is not the only measure of "strength" of a vector in $\mathbb{C}^N$

DEFINITION

The p-**norm** of a vector $x \in \mathbb{C}^N$ is given by

$$\|x\|_p = \left(\sum_{n=0}^{N-1} |x[n]|^p \right)^{1/p}$$

DEFINITION

The 1-**norm** of a vector $x \in \mathbb{C}^N$ is given by

$$\|x\|_1 = \sum_{n=0}^{N-1} |x[n]|$$

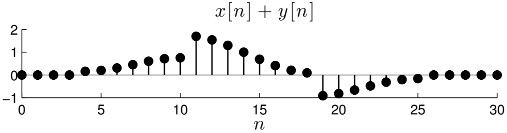
Strength of a Vector: ∞-Norm

DEFINITION

The ∞-**norm** of a vector $x \in \mathbb{C}^N$ is given by

$$\|x\|_\infty = \max_n |x[n]|$$

- $\|x\|_\infty$ is simply the largest entry in the vector x (in absolute value)

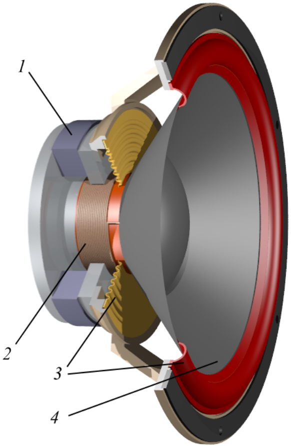


- While $\|x\|_2^2$ measures the energy in a signal, $\|x\|_\infty$ measures the peak value (of the magnitude); both are very useful in applications

- Interesting mathematical fact: $\|x\|_\infty = \lim_{p\to\infty} \|x\|_p$

Physical Significance of Norms (1)

- Two norms have special physical significance
  - $\|x\|_2^2$: energy in x
  - $\|x\|_\infty$: peak value in x
- A **loudspeaker** is a transducer that converts electrical signals into acoustic signals
- Conventional loudspeakers consist of a paper cone (4) that is joined to a coil of wire (2) that is wound around a permanent magnet (1)
- If the energy $\|x\|_2^2$ is too large, then the coil of wire will melt from excessive heating
- If the peak value $\|x\|_\infty$ is too large, then the large back and forth excursion of the coil of wire will tear it off of the paper cone



Physical Significance of Norms (2)

- Consider a **robotic car** that we wish to guide down a roadway
- How to measure the amount of deviation from the center of the driving lane?
- Let x be a vector of measurements of the car's GPS position and let y be a vector containing the GPS positions of the center of the driving lane
- Clearly we would like to make the error signal $y - x$ "small"; but how to measure smallness?
- Minimizing $\|y - x\|_2^2$, energy in the error signal, will tolerate a few large deviations from the lane center (not very safe)
- Minimizing $\|y - x\|_\infty$, the maximum of the error signal, will not tolerate any large deviations from the lane center (much safer)

Normalizing a Vector

DEFINITION

A vector x is **normalized** (in the 2-norm) if $\|x\|_2 = 1$

- Normalizing a vector is easy; just scale it by $\frac{1}{\|x\|_2}$
- Ex: $x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \qquad ||x||_2 = \sqrt{\sum_{n=0}^{N-1} |x[n]|^2} = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$

$$x' = \frac{1}{\sqrt{14}} x = \frac{1}{\sqrt{14}} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{14} \\ 2/\sqrt{14} \\ 3/\sqrt{14} \end{bmatrix}, \qquad ||x'||_2 = 1$$

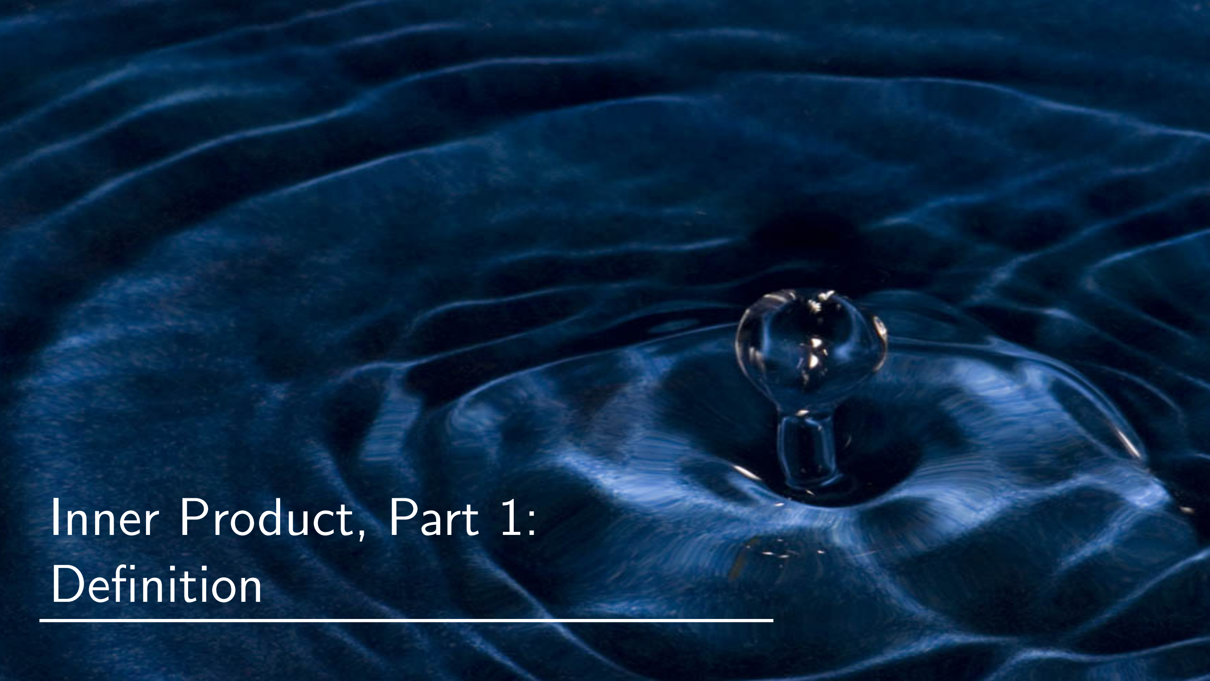
Summary

- Linear algebra provides power tools to study signals and systems
- Signals are **vectors** that live in a vector space
- In this lecture, we studied the vector spaces $\mathbb{R}^N$ and $\mathbb{C}^N$
- We can combine several signals to form one new signal via a **linear combination**
- Linear combination is basically a matrix/vector multiplication
- Norms measure the “strength” of a signal; we introduced the 2- 1-, and ∞-norms

Inner Product

Table of Contents

- Lecture in three parts:
  - Part 1: Inner Product Definition
  - Part 2: Harmonic Sinusoids are Orthogonal + Matlab Demo
  - Part 3: Matrix Multiplication and Inner Product



Inner Product, Part 1: Definition

The Geometry of Signals

- Up to this point, we have developed the viewpoint of “signals as vectors” in a vector space
- We have focused on quantities related to individual vectors, ex: norm (strength)
- Now we turn to quantities related to pairs of vectors, inner product
- A powerful and ubiquitous signal processing tool

Aside: Transpose of a Vector

- Recall that the **transpose** operation ^{T} converts a column vector to a row vector (and vice versa)

$$\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}^T = \begin{bmatrix} x[0] & x[1] & \cdots & x[N-1] \end{bmatrix}$$

- In addition to transposition, the **conjugate transpose** (aka Hermitian transpose) operation ^{H} takes the complex conjugate

$$\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}^H = \begin{bmatrix} x[0]^* & x[1]^* & \cdots & x[N-1]^* \end{bmatrix}$$

Inner Product

DEFINITION

The **inner product** (or dot product) between two vectors $x, y \in \mathbb{C}^N$ is given by

$$\langle x, y \rangle = y^H x = \sum_{n=0}^{N-1} x[n]\, y[n]^*$$

- The inner product takes two signals (vectors in $\mathbb{C}^N$) and produces a single (complex) number
- **Angle** between two vectors $x, y \in \mathbb{R}^N$

$$\cos\theta_{x,y} = \frac{\langle x, y \rangle}{\|x\|_2\, \|y\|_2}$$

- **Angle** between two vectors $x, y \in \mathbb{C}^N$

$$\cos\theta_{x,y} = \frac{\mathrm{Re}\{\langle x, y \rangle\}}{\|x\|_2\, \|y\|_2}$$

Inner Product Example 1

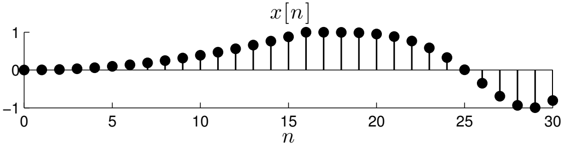
- Consider two vectors in $\mathbb{R}^2$: $x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad y = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$

- $\|x\|_2^2 = 1^2 + 2^2 = 5, \qquad \|y\|_2^2 = 3^2 + 2^2 = 13$

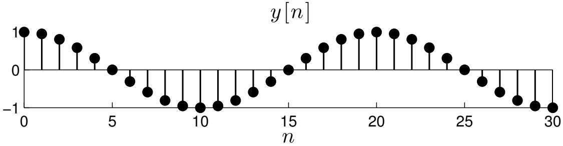
- $\theta_{x,y} = \arccos\left(\frac{1\times 3\ +\ 2\times 2}{\sqrt{5}\sqrt{13}}\right) = \arccos\left(\frac{7}{\sqrt{65}}\right) \approx 0.519 \text{ rad} \approx 29.7^\circ$

Inner Product Example 2

- Signal x



- Signal y



- Inner product computed using Matlab: $\langle x, y \rangle = y^T x = 5.995$

- Angle computed using Matlab: $\theta_{x,y} = 64.9°$

2-Norm from Inner Product

- Question: What's the inner product of a signal with itself?

$$\langle x, x\rangle = \sum_{n=0}^{N-1} x[n]\, x[n]^* = \sum_{n=0}^{N-1} |x[n]|^2 = \|x\|_2^2$$

- Answer: The 2-norm!

- Mathematical aside: This property makes the 2-norm very special; no other p-norm can be computed via the inner product like this

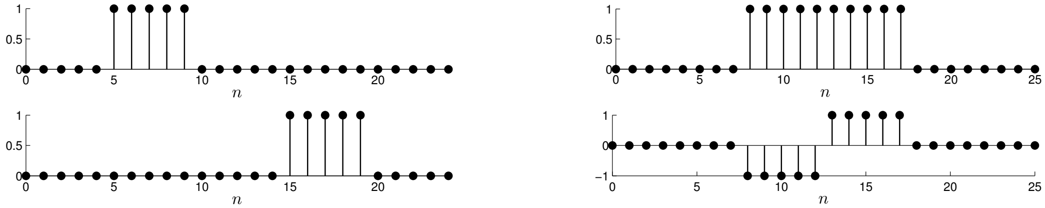
Orthogonal Vectors

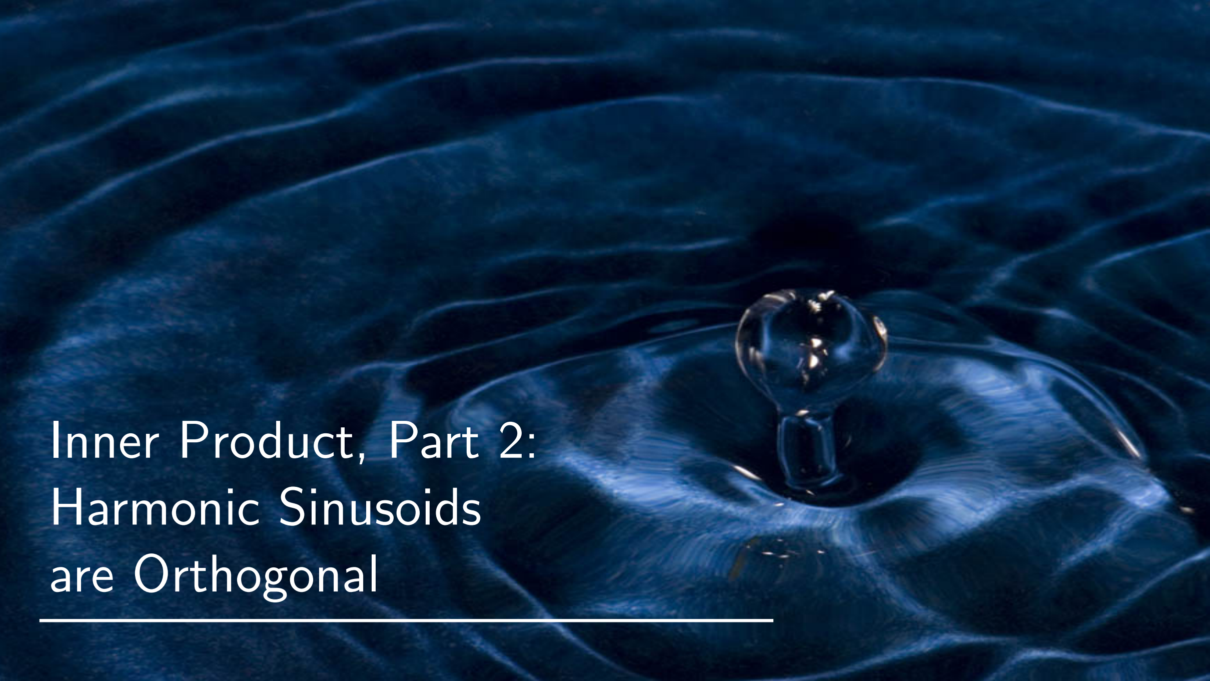
DEFINITION

Two vectors $x, y \in \mathbb{C}^N$ are **orthogonal** if

$$\langle x, y \rangle = 0$$

- $\langle x, y \rangle = 0 \;\Rightarrow\; \theta_{x,y} = \pi \text{ rad} = 90^\circ$
- Ex: Two sets of orthogonal signals





Inner Product, Part 2: Harmonic Sinusoids are Orthogonal

Harmonic Sinusoids are Orthogonal

$$d_k[n] = e^{j\frac{2\pi k}{N}n}, \quad n, k, N \in \mathbb{Z}, \;\; 0 \le n \le N-1, \;\; 0 \le k \le N-1$$

- Claim: $\langle d_k | d_l \rangle = 0, \quad k \neq l$ (a key result for the DFT)
- Verify by direct calculation

$$\begin{aligned}
\langle d_k | d_l \rangle &= \sum_{n=0}^{N-1} d_k[n]\, d_l^*[n] \;=\; \sum_{n=0}^{N-1} e^{j\frac{2\pi k}{N}n} \,(e^{j\frac{2\pi l}{N}n})^* \;=\; \sum_{n=0}^{N-1} e^{j\frac{2\pi k}{N}n}\, e^{-j\frac{2\pi l}{N}n} \\
&= \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(k-l)n} \quad \text{let } r = k - l \in \mathbb{Z}, r \neq 0 \\
&= \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}rn} \;=\; \sum_{n=0}^{N-1} a^n \quad \text{with } a = e^{j\frac{2\pi}{N}r}, \text{then use } \sum_{n=0}^{N-1} a^n = \frac{1-a^N}{1-a} \\
&= \frac{1 - e^{j\frac{2\pi r N}{N}}}{1 - e^{j\frac{2\pi r}{N}}} \;=\; 0 \;\; \checkmark
\end{aligned}$$

Normalizing Harmonic Sinusoids

$$d_k[n] = e^{j\frac{2\pi k}{N}n}, \quad n, k, N \in \mathbb{Z}, \ \ 0 \leq n \leq N-1, \ \ 0 \leq k \leq N-1$$

- Claim: $\|d_k\|_2 = \sqrt{N}$
- Verify by direct calculation

$$\|d_k\|_2^2 \ = \ \sum_{n=0}^{N-1} |d_k[n]|^2 \ = \ \sum_{n=0}^{N-1} |e^{j\frac{2\pi k}{N}n}|^2 \ = \ \sum_{n=0}^{N-1} 1 \ = \ N \quad \checkmark$$

- **Normalized harmonic sinusoids**

$$\widetilde{d}_k[n] = \frac{1}{\sqrt{N}}{}^{j\frac{2\pi k}{N}n}, \quad n, k, N \in \mathbb{Z}, \ \ 0 \leq n \leq N-1, \ \ 0 \leq k \leq N-1$$

Inner Product, Part 3: Matrix Multiplication and Inner Product

Recall: Matrix Multiplication as a Linear Combination of Columns

- Consider the matrix multiplication $y = Xa$

- The row-n, column-m element of the $N \times M$ matrix $[X]_{n,m} = x_m[n]$

- We can compute y as a **linear combination** of the **columns of** X weighted by the elements in a

$$y = \begin{bmatrix} \vdots \\ y[n] \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ x_0[n] & x_1[n] & \cdots & x_{M-1}[n] \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{M-1} \end{bmatrix} = Xa$$

- Sum-based formula for $y[n]$

$$y[n] = \sum_{m=0}^{M-1} \alpha_m \, x_m[n], = \sum_{m=0}^{M-1} \alpha_m \, (\text{column } m \text{ of } X), \quad 0 \leq n \leq N-1$$

Matrix Multiplication as a Sequence of Inner Products of Rows

- Consider the matrix multiplication $\quad y = Xa$

- The row-n, column-m element of the $N \times M$ matrix $[X]_{n,m} = x_m[n]$

- We can compute each element $y[n]$ in y as the **inner product** of the n**-th row of** X with the vector a (true for $\mathbb{R}^N$; need to take a * into account in $\mathbb{C}^N$)

$$y \;=\; \begin{bmatrix} \vdots \\ y[n] \\ \vdots \end{bmatrix} \;=\; \begin{bmatrix} \vdots & \vdots & & \vdots \\ x_0[n] & x_1[n] & \cdots & x_{M-1}[n] \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{M-1} \end{bmatrix} \;=\; Xa$$

- Can write $y[n]$

$$y[n] \;=\; \sum_{m=0}^{M-1} \alpha_m \, x_m[n] \;=\; \langle \text{row } n \text{ of } X,\, a \rangle, \quad 0 \le n \le N-1$$

Summary

- **Inner product** measures the similarity between two signals

$$\langle x, y \rangle = y^H x = \sum_{n=0}^{N-1} x[n]\, y[n]^*$$

- Angle between two signals

$$\cos \theta_{x,y} = \frac{\mathrm{Re}\{\langle x, y \rangle\}}{\|x\|_2 \, \|y\|_2}$$

Cauchy Schwarz Inequality

Comparing Signals

- Inner product and angle between vectors enable us to compare signals

$$\langle x, y\rangle = y^H x = \sum_{n=0}^{N-1} x[n]\, y[n]^*$$

$$\cos\theta_{x,y} = \frac{\mathrm{Re}\{\langle x, y\rangle\}}{\|x\|_2\, \|y\|_2}$$

- The Cauchy Schwarz Inequality quantifies the comparison

- A powerful and ubiquitous signal processing tool

- Note: Our development will emphasize intuition over rigor

Cauchy-Schwarz Inequality (1)

- Focus on real-valued signals in $\mathbb{R}^N$ (the extension to $\mathbb{C}^N$ is easy)

- Recall that $\cos\theta_{x,y} = \frac{\langle x,y\rangle}{\|x\|_2\,\|y\|_2}$

- Now, use the fact that $0 \leq |\cos\theta| \leq 1$ to write

$$0 \;\leq\; \left|\frac{\langle x,y\rangle}{\|x\|_2\,\|y\|_2}\right| \;\leq\; 1$$

- Rewrite as the **Cauchy-Schwarz Inequality** (CSI)

$$0 \;\leq\; |\langle x,y\rangle| \;\leq\; \|x\|_2\,\|y\|_2$$

- Interpretation: The inner product $\langle x,y\rangle$ measures the **similarity** of x to y

Cauchy-Schwarz Inequality (2)

$$0 \;\leq\; |\langle x, y \rangle| \;\leq\; \|x\|_2 \, \|y\|_2$$

- Interpretation: The inner product $\langle x, y \rangle$ measures the **similarity** of x to y

- Two extreme cases:
  - Lower bound: $\langle x, y \rangle = 0$ or $\theta_{x,y} = 90^\circ$: x and y are most different when they are orthogonal
  - Upper bound: $\langle x, y \rangle = \|x\|_2 \|y\|_2$ or $\theta_{x,y} = 0^\circ$: x and y are most similar when they are collinear (aka linearly dependent, $y = \alpha x$)

- It is hard to understate the importance and ubiquity of the CSI!

Cauchy-Schwarz Inequality Applications

- How does a digital communication system decide whether the signal corresponding to a "0" was transmitted or the signal corresponding to a "1"? (Hint: CSI)
- How does a radar or sonar system find targets in the signal it receives after transmitting a pulse? (Hint: CSI)
- How does many computer vision systems find faces in images? (Hint: CSI)

Summary

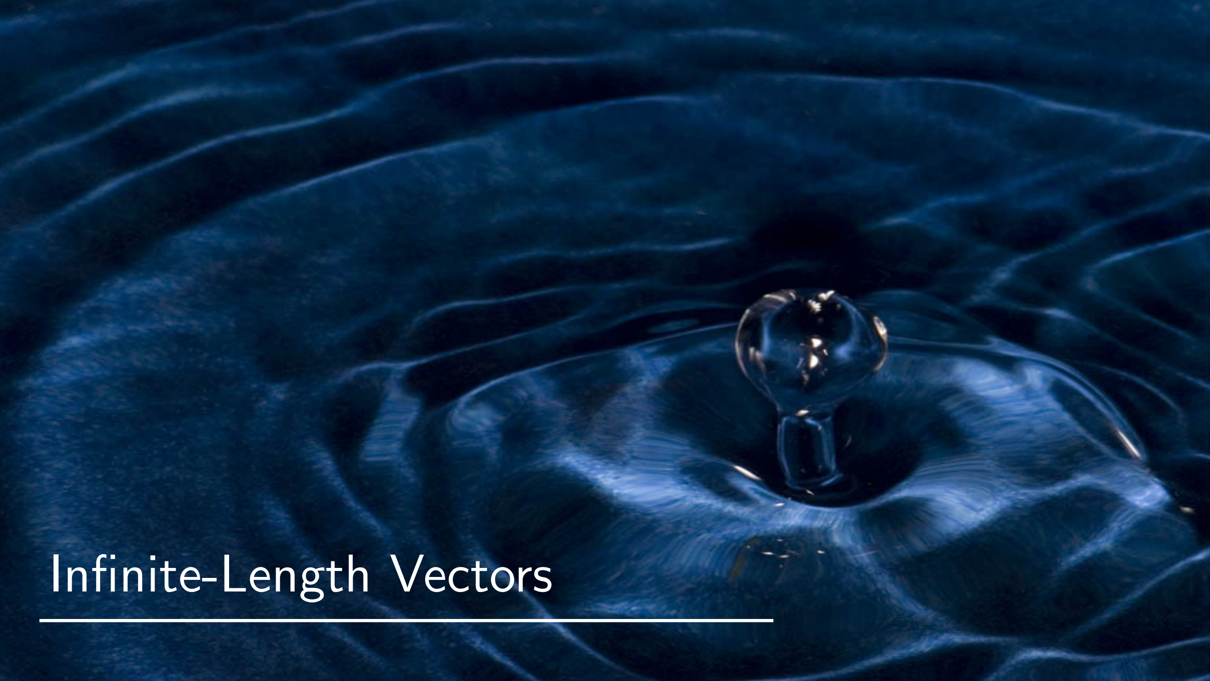
- **Inner product** measures the similarity between two signals

$$\langle x, y\rangle = y^H x = \sum_{n=0}^{N-1} x[n]\, y[n]^*$$

- **Cauchy-Schwarz Inequality** (CSI)

$$0 \leq \left| \frac{\langle x, y\rangle}{\|x\|_2\, \|y\|_2} \right| \leq 1$$

  - Similar signals – close to upper bound (1)
  - Different signals – close to lower bound (0)

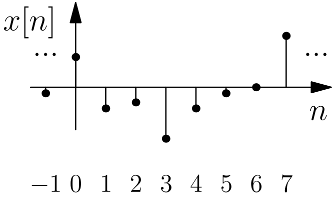


Infinite-Length Vectors

From Finite to Infinite-Length Vectors

- Up to this point, we have developed some useful tools for dealing with finite-length vectors (signals) that live in $\mathbb{R}^N$ or $\mathbb{C}^N$: Norms, Inner product, Linear combination
- It turns out that these tools can be generalized to infinite-length vectors (signals) by letting $N \to \infty$ (infinite-dimensional vector space, aka Hilbert Space)

$$x[n], \quad \infty < n < \infty, \qquad x = \begin{bmatrix} \vdots \\ x[-2] \\ x[-1] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix}$$



- Obviously such a signal cannot be loaded into Matlab; however this viewpoint is still useful in many situations
- We will spell out the generalizations with emphasis on what changes from the finite-length case

2-Norm of an Infinite-Length Vector

DEFINITION

The 2-**norm** of an infinite-length vector x is given by

$$\|x\|_2 = \sqrt{\sum_{n=-\infty}^{\infty} |x[n]|^2}$$

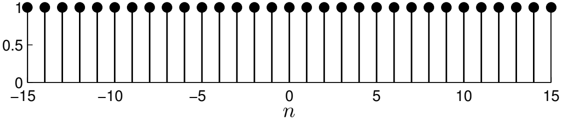
The **energy** of x is given by $(\|x\|_2)^2 = \|x\|_2^2$

- When it is clear from context, we will suppress the subscript “2” in $\|x\|_2$ and just write $\|x\|$

- What changes from the finite-length case: Not every infinite-length vector has a finite 2-norm

ℓ_2 Norm of an Infinite-Length Vector – Example

- Signal: $x[n] = 1, \quad \infty < n < \infty$



- 2-norm:

$$\|x\|_2^2 \;=\; \sum_{n=-\infty}^{\infty} |x[n]|^2 \;=\; \sum_{n=-\infty}^{\infty} 1 \;=\; \infty$$

- Infinite energy!

p- and 1-Norms of an Infinite-Length Vector

DEFINITION

The p-**norm** of an infinite-length vector x is given by

$$\|x\|_p = \left(\sum_{n=-\infty}^{\infty} |x[n]|^p \right)^{1/p}$$

DEFINITION

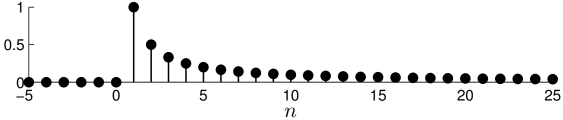
The 1-**norm** of an infinite-length vector x is given by

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

- What changes from the finite-length case: Not every infinite-length vector has a finite p-norm

1- and 2-Norms of an Infinite-Length Vector – Example

- Signal: $x[n] = \begin{cases} 0 & n \leq 0 \\ \frac{1}{n} & n \geq 1 \end{cases}$



- 1-norm

$$\|x\|_1 \;=\; \sum_{n=-\infty}^{\infty} |x[n]| \;=\; \sum_{n=1}^{\infty} \frac{1}{n} \;=\; \infty$$

- 2-norm

$$\|x\|_2^2 \;=\; \sum_{n=-\infty}^{\infty} |x[n]|^2 \;=\; \sum_{n=1}^{\infty} \left|\frac{1}{n}\right|^2 \;=\; \sum_{n=1}^{\infty} \frac{1}{n^2} \;=\; \frac{\pi^2}{6} \approx 1.64 \;<\; \infty$$

∞-Norm of an Infinite-Length Vector

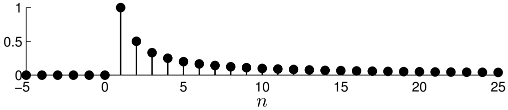
DEFINITION

The ∞-**norm** of an infinite-length vector x is given by

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

- What changes from the finite-length case: "sup" is a generalization of $\max$ to infinite-length signals that lies beyond the scope of this course

- In both of the above examples, $\|x\|_\infty = 1$



Inner Product of Infinite-Length Signals

DEFINITION

The **inner product** between two infinite-length vectors x, y is given by

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

- The inner product takes two signals and produces a single (complex) number

- **Angle** between two real-valued signals

$$\cos\theta_{x,y} = \frac{\langle x, y \rangle}{\|x\|_2 \, \|y\|_2}$$

- **Angle** between two complex-valued signals

$$\cos\theta_{x,y} = \frac{\mathrm{Re}\{\langle x, y \rangle\}}{\|x\|_2 \, \|y\|_2}$$

Linear Combination of Infinite-Length Vectors

- The concept of a linear combination extends to infinite-length vectors

- What changes from the finite-length case: We will be especially interested in linear combinations of infinitely many infinite-length vectors

$$y = \sum_{m=-\infty}^{\infty} \alpha_m x_m$$

Linear Combination = Infinite Matrix Multiplication

- Step 1: Stack the vectors x_m as column vectors into a "matrix" with infinitely many rows and columns

$$X = \big[\cdots | x_{-1} | x_0 | x_1 | \cdots \big]$$

- Step 2: Stack the scalars α_m into an infinitely tall column vector $a = \begin{bmatrix} \vdots \\ \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \\ \vdots \end{bmatrix}$

- Step 3: We can now write a linear combination as the matrix/vector product

$$y \;=\; \sum_{m=-\infty}^{\infty} \alpha_m x_m \;=\; \big[\cdots | x_{-1} | x_0 | x_1 | \cdots \big] \begin{bmatrix} \vdots \\ \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \\ \vdots \end{bmatrix} \;=\; X a$$

Linear Combination = Infinite Matrix Multiplication (The Gory Details)

- Vectors: $x_m = \begin{bmatrix} \vdots \\ x_m[-1] \\ x_m[0] \\ x_m[1] \\ \vdots \end{bmatrix}, \quad -\infty < m < \infty,$ and Scalars: $a = \begin{bmatrix} \vdots \\ \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \\ \vdots \end{bmatrix}$

- Infinite matrix: $X = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & x_{-1}[-1] & x_0[-1] & x_1[-1] & \cdots \\ \cdots & x_{-1}[0] & x_0[0] & x_1[0] & \cdots \\ \cdots & x_{-1}[1] & x_0[1] & x_1[1] & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$

- Note: The row-n, column-m element of the matrix $[X]_{n,m} = x_m[n]$

- Linear combination $= Xa$

Linear Combination = Infinite Matrix Multiplication (Summary)

- Linear combination $y = Xa$

- The row-n, column-m element of the infinitely large matrix $[X]_{n,m} = x_m[n]$

$$y = \begin{bmatrix} \vdots \\ y[n] \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & x_m[n] & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \alpha_m \\ \vdots \end{bmatrix} = Xa$$

- Sum-based formula for $y[n]$

$$y[n] = \sum_{m=-\infty}^{\infty} \alpha_m \, x_m[n]$$

Summary

- Linear algebra concepts like norm, inner product, and linear combination work just as well with infinite-length signals as with finite-length signals

- Only a few changes from the finite-length case
  - Not every infinite-length vector has a finite 1-, 2-, or ∞-norm
  - Linear combinations can involve infinitely many vectors

Acknowledgements