ENGR 76

Information Science and Engineering

Lecture 7: Frequency Domain Representation III

Siddharth Chandak

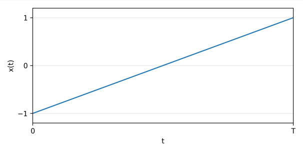
Recap

Finite-duration Signals

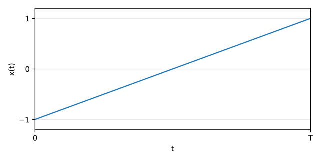
- Suppose we are given a signal of duration T seconds
- How to represent the signal using sines and cosines?

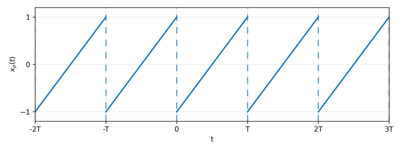
Periodic Extension

- Signal $x(t)$ defined from $t = 0$ to $t = T$
- **Periodic Extension:** $x_p(t)$ is obtained by repeating $x(t)$ again and again in $[T, 2T]$, $[2T, 3T]$, $[-T, 0]$, $[-2T, -T]$...



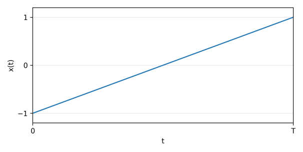
Periodic Extension



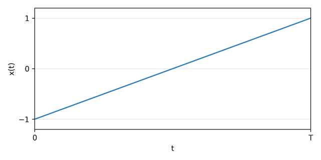


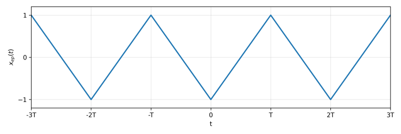
Even Periodic Extension

- Signal $x(t)$ defined from $t = 0$ to $t = T$
- **Even Periodic Extension** $x_{ep}(t)$:
  - First mirror $x(t)$ about y-axis to get its image from $-T$ to 0
  - Then repeat it again and again in $[T, 3T]$, $[-3T, -T]$...



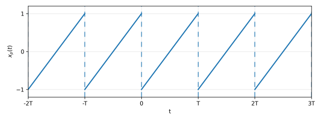
Even Periodic Extension

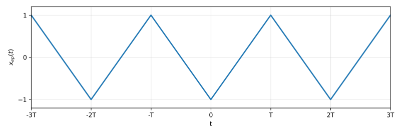




Periodic and Even Periodic Extension

Which of these is desirable for Fourier representation?





Even Periodic Extensions

- If our original finite-duration signal is continuous in $[0, T]$, then its even periodic extension will also be continuous
- Periodic extension can have jump discontinuities
- Even periodic extension is desirable
    - Requires fewer terms in its Fourier representation

Fourier Series for Finite-Duration Signal

- Given a signal $x(t)$ defined in $[0, T]$
- Fourier Representation of $x(t)$ for $t \in [0, T]$:

$$x(t) = b_0 + \sum_{j=1}^{\infty} b_j \cos\left(2\pi \frac{j}{2T} t\right)$$

  - Note the $2T$ in $\frac{j}{2T}$: period of the even periodic extension is $2T$

Discrete Time Signals

Discrete Time Signals

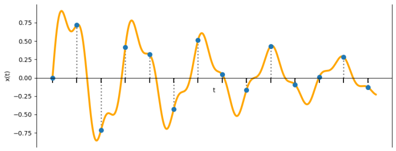
- Consider the array of length L

$$y[0], y[1], \ldots, y[L-1]$$

- Discrete time index $n = 0, 1, 2, \ldots, L-1$
- Usual Notation:
  - (t): continuous time t
  - $[n]$: discrete time n

Why discrete time signals?

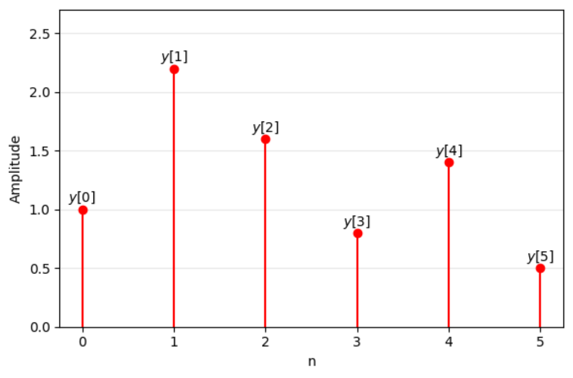
- We do not store a continuum
- Sampling an underlying continuous time signal
    - Audio: sampling an acoustic signal in time
    - Image: sampling the intensities in space (pixels)



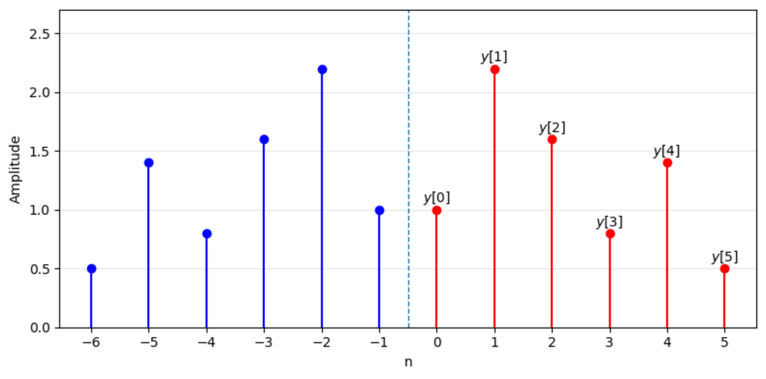
- What intervals to sample at? How to reconstruct the continuous time signal from the samples? — **Next Module**

Discrete Time Signals

- How can we represent a discrete time signal using cosines?
  - Same intuition as even periodic extension

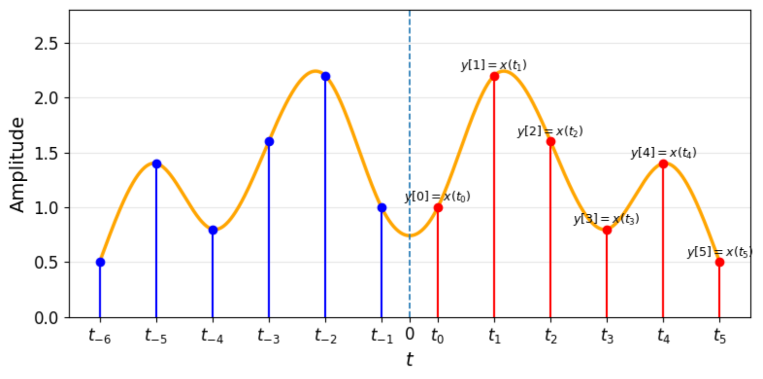


Mirror Image



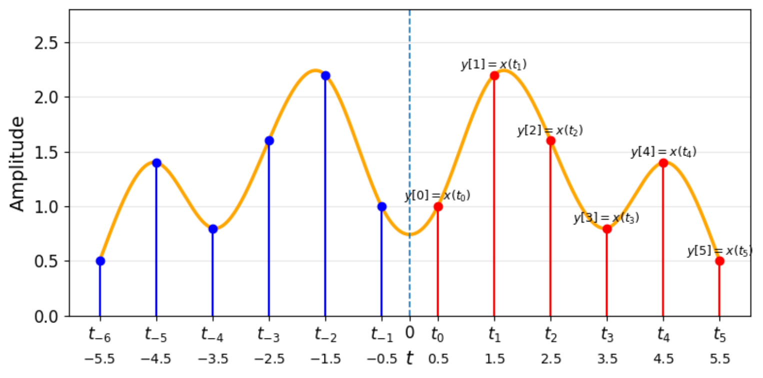
Continuous Time Interpretation

- Some underlying continuous time signal
- discrete time index $n \leftrightarrow$ continuous time index t_n
- discrete time signal $y[n] \leftrightarrow$ continuous time signal samples $x(t_n)$
- $t_n = ?$
  - Want intervals of length 1, i.e., $t_i - t_{i-1} = 1$
  - Want $t_0 = -t_{-1}$ (symmetry)



Continuous Time Interpretation

- discrete time index $n \leftrightarrow$ continuous time index t_n
- $t_n = n + 0.5$



Continuous Time Interpretation

- Recall the Fourier cosine series:

$$x(t) = b_0 + \sum_{j=1}^{\infty} \cos\left(2\pi \frac{j}{2T} t\right)$$

- Intuitive connection to discrete time:
  - Length $T \leftrightarrow L$
  - Sample at $t_n = n + 0.5$ to get $x(t_n)$ which is interpreted as $y[n]$

$$\begin{aligned} y[n] &= b_0 + \sum_{j=1}^{\infty} b_j \cos\left(2\pi \frac{j}{2L}(n + 0.5)\right) \\ &= b_0 + \sum_{j=1}^{\infty} b_j \cos\left(\frac{\pi \cdot j \cdot (n + 0.5)}{L}\right) \end{aligned}$$

L Equations ($L = 4$)

$$y[0] = b_0 \times 1 + b_1 \times \cos\left(\frac{\pi \cdot 1 \cdot 0.5}{4}\right) + b_2 \times \cos\left(\frac{\pi \cdot 2 \cdot 0.5}{4}\right) + b_3 \times \cos\left(\frac{\pi \cdot 3 \cdot 0.5}{4}\right) + \dots$$

$$y[1] = b_0 \times 1 + b_1 \times \cos\left(\frac{\pi \cdot 1 \cdot 1.5}{4}\right) + b_2 \times \cos\left(\frac{\pi \cdot 2 \cdot 1.5}{4}\right) + b_3 \times \cos\left(\frac{\pi \cdot 3 \cdot 1.5}{4}\right) + \dots$$

$$y[2] = b_0 \times 1 + b_1 \times \cos\left(\frac{\pi \cdot 1 \cdot 2.5}{4}\right) + b_2 \times \cos\left(\frac{\pi \cdot 2 \cdot 2.5}{4}\right) + b_3 \times \cos\left(\frac{\pi \cdot 3 \cdot 2.5}{4}\right) + \dots$$

$$y[3] = b_0 \times 1 + b_1 \times \cos\left(\frac{\pi \cdot 1 \cdot 3.5}{4}\right) + b_2 \times \cos\left(\frac{\pi \cdot 2 \cdot 3.5}{4}\right) + b_3 \times \cos\left(\frac{\pi \cdot 3 \cdot 3.5}{4}\right) + \dots$$

Do we need the additional terms $\dots$ here?

Do we need summation till infinity?

- We have been given the vector $(y[0], y[1], \ldots, y[L-1])$ of length L
- L degrees of freedom or L pieces of information
- We are representing these using the coefficients $b_0, b_1, b_2, \ldots$
    - How many coefficients/terms do we need?
- Alternatively, we have L equations - how many unknowns $b_0, b_1 \ldots$ can we find?

Discrete Cosine Transform

Fact

Given discrete time signal $y[0], \ldots, y[L-1]$, then $y[n]$ for $n = 0, \ldots, L-1$ can be represented as

$$y[n] = b_0 + \sum_{j=1}^{L-1} b_j \cos\left(\frac{\pi \cdot j \cdot (n+0.5)}{L}\right).$$

Moreover, the coefficients $b_0, \ldots, b_{L-1}$ are unique for given signal.

Representations

- Time-domain representation: $y[0], y[1], \ldots, y[L-1]$
  - Signal at different time indices
- Frequency-domain representation: $b_0, b_1, \ldots, b_{L-1}$
  - Coefficients

Interpretations

$$y[0] = b_0 \times 1 + b_1 \times \cos\left(\frac{\pi \cdot 1 \cdot 0.5}{4}\right) + b_2 \times \cos\left(\frac{\pi \cdot 2 \cdot 0.5}{4}\right) + b_3 \times \cos\left(\frac{\pi \cdot 3 \cdot 0.5}{4}\right)$$

$$y[1] = b_0 \times 1 + b_1 \times \cos\left(\frac{\pi \cdot 1 \cdot 1.5}{4}\right) + b_2 \times \cos\left(\frac{\pi \cdot 2 \cdot 1.5}{4}\right) + b_3 \times \cos\left(\frac{\pi \cdot 3 \cdot 1.5}{4}\right)$$

$$y[2] = b_0 \times 1 + b_1 \times \cos\left(\frac{\pi \cdot 1 \cdot 2.5}{4}\right) + b_2 \times \cos\left(\frac{\pi \cdot 2 \cdot 2.5}{4}\right) + b_3 \times \cos\left(\frac{\pi \cdot 3 \cdot 2.5}{4}\right)$$

$$y[3] = b_0 \times 1 + b_1 \times \cos\left(\frac{\pi \cdot 1 \cdot 3.5}{4}\right) + b_2 \times \cos\left(\frac{\pi \cdot 2 \cdot 3.5}{4}\right) + b_3 \times \cos\left(\frac{\pi \cdot 3 \cdot 3.5}{4}\right)$$

Stack it!

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \end{bmatrix} = b_0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + b_1 \begin{bmatrix} \cos\left(\frac{\pi \cdot 1 \cdot 0.5}{4}\right) \\ \cos\left(\frac{\pi \cdot 1 \cdot 1.5}{4}\right) \\ \cos\left(\frac{\pi \cdot 1 \cdot 2.5}{4}\right) \\ \cos\left(\frac{\pi \cdot 1 \cdot 3.5}{4}\right) \end{bmatrix} + b_2 \begin{bmatrix} \cos\left(\frac{\pi \cdot 2 \cdot 0.5}{4}\right) \\ \cos\left(\frac{\pi \cdot 2 \cdot 1.5}{4}\right) \\ \cos\left(\frac{\pi \cdot 2 \cdot 2.5}{4}\right) \\ \cos\left(\frac{\pi \cdot 2 \cdot 3.5}{4}\right) \end{bmatrix} + b_3 \begin{bmatrix} \cos\left(\frac{\pi \cdot 3 \cdot 0.5}{4}\right) \\ \cos\left(\frac{\pi \cdot 3 \cdot 1.5}{4}\right) \\ \cos\left(\frac{\pi \cdot 3 \cdot 2.5}{4}\right) \\ \cos\left(\frac{\pi \cdot 3 \cdot 3.5}{4}\right) \end{bmatrix}.$$

Basis Functions

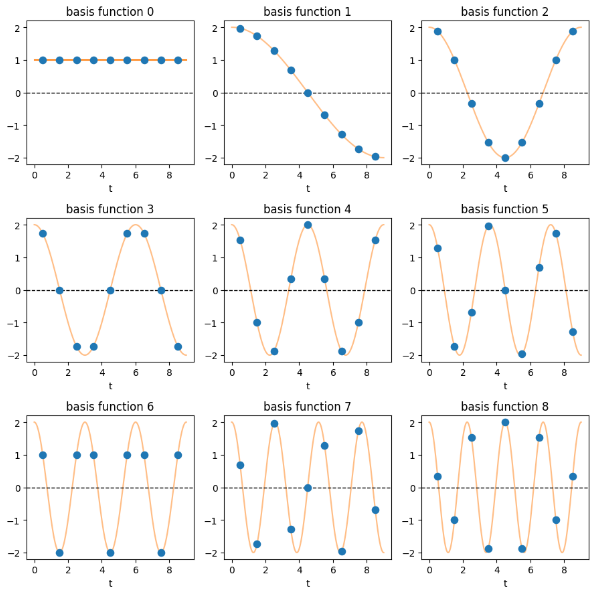
- We define each of these stacked vectors as “basis functions”
- ϕ_j denotes the j-th cosine basis function such that
    - $\phi_0[n] = 1$ for all n
    - $\phi_j[n] = \cos\left(\frac{\pi \cdot j \cdot (n+0.5)}{L}\right)$ for $j \neq 0$

$$\phi_0 = [1, 1, \ldots, 1]$$

$$\phi_j = \left[\cos\left(\frac{\pi \cdot j \cdot (0.5)}{L}\right), \cos\left(\frac{\pi \cdot j \cdot (1.5)}{L}\right), \ldots, \cos\left(\frac{\pi \cdot j \cdot (L-1+0.5)}{L}\right)\right]$$

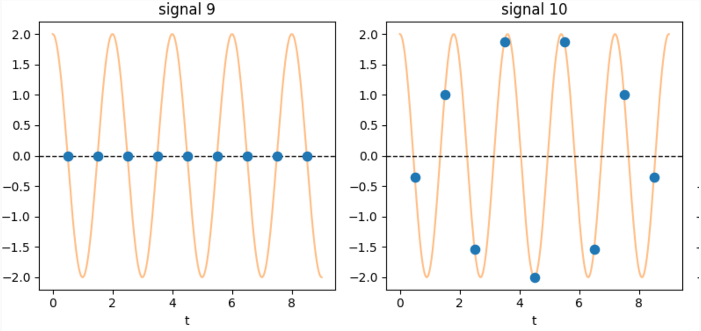
- Each ϕ_j is a discrete time signal
    - Samples of the cosine with frequency $\frac{j}{2L}$ at times $[n + 0.5]$

What do these look like? $(L = 9)$



What happens if we add more?

- We decided that only the first L (0 to $L-1$) terms are needed
- How do the next terms look like? Are they useful?



Basis Interpretation of DCT

$$\overline{y} = \begin{bmatrix} y[0] \\ \vdots \\ y[L-1] \end{bmatrix} \text{ and } \phi_j = \begin{bmatrix} \phi_j[0] \\ \vdots \\ \phi_j[L-1] \end{bmatrix}$$

Fact

The vectors $\phi_0, \ldots, \phi_{L-1}$ act as a basis for $\mathbb{R}^L$, i.e., any signal (vector) $\overline{y}$ of length L can be represented as

$$\overline{y} = \sum_{j=0}^{L-1} b_j \phi_j.$$

Moreover, the coefficients $b_0, \ldots, b_{L-1}$ are unique.

- Linear combination of all these basis functions gives desired signal

$$y[n] = \sum_{j=0}^{L-1} b_j \phi_j[n]$$

Computing Coefficients

$$
\begin{aligned}
b_0 &= \frac{1}{L}\sum_{n=0}^{L-1} y[n] \\
b_j &= \frac{2}{L}\sum_{n=0}^{L-1} y[n]\cos\left(\frac{\pi \cdot j \cdot (n+0.5)}{L}\right) \\
&= \frac{2}{L}\sum_{n=0}^{L-1} y[n]\phi_j[n]
\end{aligned}
$$

- b_0 just denotes the average of the signal

Matrix Representation

- Clean way to represent the transform
- $L \times L$ matrix $\Phi = [\phi_0 \ \phi_1 \ \ldots \ \phi_{L-1}]$ - stacking cosine basis functions as columns
- Inverse DCT (IDCT) — from frequency-domain to time-domain:

$$\begin{bmatrix} y[0] \\ \vdots \\ y[L-1] \end{bmatrix} = \Phi \begin{bmatrix} b_0 \\ \vdots \\ b_{L-1} \end{bmatrix}$$

- DCT — from time-domain to frequency domain:

$$\begin{bmatrix} b_0 \\ \vdots \\ b_{L-1} \end{bmatrix} = \Phi^{-1} \begin{bmatrix} y[0] \\ \vdots \\ y[L-1] \end{bmatrix}$$

- **Additional (optional)** information about the matrix Φ and linear algebra intuition - presented in lecture notes

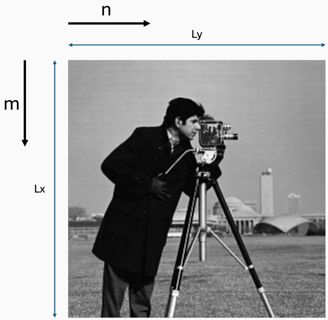
Computational Complexity

- Multiplying an $L \times L$ matrix with a vector of dimension L requires $\mathcal{O}(L^2)$ computations
- But the DCT transform has additional structure
  - Can be done using $\mathcal{O}(L \log L)$ computations
  - **Fast Fourier Transform**
  - Efficient algorithm

2-D DCT

2-D Signals (Images)

- $X[m,n]$ for $m \in \{0,1,\dots,L_x-1\}, n \in \{0,1,\dots,L_y-1\}$
- Convention:



- $X[m,n]$ is the pixel at m-th row and n-th column

2-D Discrete Cosine Transform

- 2-D DCT:

$$X[m,n] = \sum_{i=0}^{L_x-1} \sum_{j=0}^{L_y-1} A[i,j]\phi_{i,j}[m,n]$$

- Basis functions:

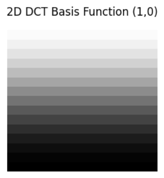
$$\phi_{i,j}[m,n] = \cos\left(\frac{\pi \cdot i \cdot (m+0.5)}{L_x}\right)\cos\left(\frac{\pi \cdot j \cdot (n+0.5)}{L_y}\right)$$

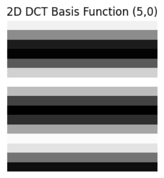
- Each basis function is an image now
- Desired image X can be obtained using linear combination of these basis functions

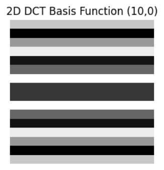
Representations

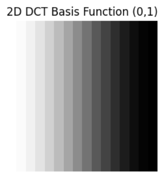
- $X[m,n]$ - $L_x \times L_y$ - image domain representation
  - Pixel intensities
- $A[i,j]$ - $L_x \times L_y$ - frequency domain representation
  - Coefficients

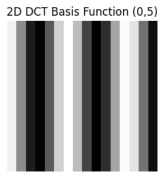
2-D DCT Basis Functions

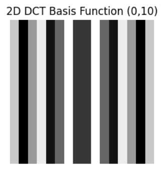


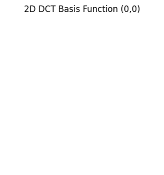


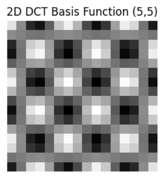


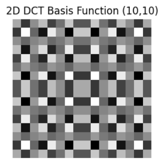












Why is 2-D DCT helpful for image compression?

- **Energy Compaction:**
  - For most natural images: DCT is sparse (concentrated in lower frequencies)
  - Most high-frequency coefficients have low magnitudes: can be discarded (i.e., low-pass filter on A)
  - Low Pass Filter on Image Visualization
- **Decorrelation:**
  - For most natural images, neighboring pixels are highly correlated
  - But in frequency-domain representation, coefficients are much less statistically dependent.
  - Huffman coding on individual symbols performs better

Thank You!