SIGNALS

CONTINUOUS/DISCRETE TIME/FREQUENCY DOMAINS
Continuous-Time Signals

• Signals in the “real world”
• Frequency content (spectrum) for finite-time signals theoretically extends to infinity
• Ref: Discrete-Time Signal Processing, Oppenheim & Schafer
Euler’s Formula

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

\[ e^{i2\pi x} = \cos(2\pi x) + i \sin(2\pi x) \]

\[ e^{-i2\pi x} = \cos(2\pi x) - i \sin(2\pi x) \]

- Called “the most remarkable formula in mathematics” by Richard Feynman [The Feynman Lectures on Physics, page 22-10]
- \( e^{ix} \) is also called a “complex sinusoid”
- \( e^{ix} \) is the kernel of the Fourier Transform
Continuous Fourier Transform

- Fourier transform pair
  - $f(x)$ is the “time domain” representation
  - $F(s)$ is the “frequency domain” representation

$$F(s) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi sx} \, dx$$

$$f(x) = \int_{-\infty}^{\infty} F(s)e^{i2\pi sx} \, ds$$
Time and Frequency Domains

• Although we defined
  – \( f(x) \) as the “time domain”
  – \( F(s) \) as the “frequency domain”
  – this is not necessarily always true

• Ex: A signal could be a function of linear distance
  – The Fourier transform could be called the “spatial frequency” (cycles per meter)

• Ex: Multi-dimensional signals
  – Signal could be an image, CAT scan or MRI or radar electromagnetic signal
Fourier Transform

- Inputs may be complex
- Outputs may be complex
- Consider the $e^{i2\pi xs}$ kernel of the transform
  - For a particular value of $s$, the kernel is a complex sinusoid of “frequency” $s$
  - All $e^{i2\pi xs}$ complex sinusoids are orthogonal
- The Fourier transform “decomposes” or “describes” a signal as a sum of complex sinusoids
Discrete-Time Signals

- Signal levels specified only at discrete time intervals ($t_{sample}$)
- Sampling frequency $= 1 / t_{sample}$
- Frequency content extends to infinity but repeats every $2\pi$ radians
  - Ex: DC content value is at $..., -4\pi, -2\pi, 0, 2\pi, 4\pi, ...$
  - Ex: Highest unambiguously-representable content value is at $..., -3\pi, -\pi, \pi, 3\pi, ...$
- Nyquist frequency $= f_{Nyq}$
  $= 1 / (2 * T_s)$
  $= (1/2) f_{sample}$
Discrete Fourier Transform (DFT)

- $X(k)$ is the DFT of the $N$-point sequence $x(n)$

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-i2\pi nk/N}, \quad k = 0, 1, \ldots, N - 1$$

- And the Inverse DFT

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{i2\pi nk/N}, \quad n = 0, 1, \ldots, N - 1$$
cos/impulse Pair

- Ref: *The Fourier Transform and Its Applications*, Ron Bracewell
- $\cos(\ )$ and its Fourier transform
- $\cos$ and $\sin$ are “pure tones” of a single frequency
cos/impulse Pair

- Example:
  - length = 1024
  - top plot: \( \cos(2\pi \times 10 \times t / \text{length}) \)
  - bottom plot: \( \text{abs}(	ext{fft}(\text{top\_waveform})) \)
**cos/impulse Pair**

- matlab code for the adjacent plots

```matlab
clear;
length = 1024;
x = 0:(length-1);

%--- cos(10)
freq = 10;
t = cos(2*pi*x*freq/length);
f = abs(fft(t));
figure(1); clf;
subplot(2,1,1);
plot(x,t);
title('Time domain');
axis([0 (length-1) -1.1 +1.1]);
grid on;
subplot(2,1,2);
plot(x,f);
title('Frequency domain');
axis([0 (length-1) 0 (length/2*1.1)]); grid on;
pdint -dtiff 1.tiff
```
cos/impulse Pair

- A higher frequency $\cos()$
impulse/sinusoid Pair

- There is a symmetry with the previous transform pairs in that an impulse in the “time” domain produces sin/cos in the “frequency” domain.
- A real one-sided impulse in the time domain results in complex sinusoids in the frequency domain.
rect/sinc Pair

- The $\text{rect}(x)$ function transforms into a $\text{sinc}(x)$ function
- $\text{rect}(x)$ is a low-pass filter
- $\text{sinc}(x) = \frac{\sin(x)}{x}$
- Can also view $\text{sinc}(x)$ as $\sin(x)$ within a $1/x$ envelope
rect/sinc Pair

- Similarly, the rect\((x)\) function in the "frequency" domain transforms into a sinc\((x)\) function in the "time" domain.

- A processor that implements a rect\((x)\) in the frequency domain is a widely-used low-pass filter (LPF).
To get a better look at the time-domain version of the LPF, we swap the two halves of the waveform using `fftshift()` in matlab, which is effectively viewing the filter’s coefficients shifted by $\pi$ to the right or left.

The sinc() shape of a filter’s time-domain coefficients is a tell-tale sign of a low-pass filter.
rect/sinc Pair

- matlab code for the rect() / sinc() plots

```matlab
clear;
length = 1024;
x = 0:(length-1);

% Time domain
x = 0:(length-1);
t = zeros(1,length);
t(1:10) = 16;
t((length-8):length) = 16;
f = fft(t); % Frequency domain

% Time domain
figure(5); clf;
subplot(2,1,1);
plot(x,t);
title('Time domain = rect()')
axis([0 (length-1) 0 +17])
grid on;

% Frequency domain
subplot(2,1,2);
plot(x,real(f),'r');
hold on;
plot(x,imag(f),'b');
title('Frequency domain = FFT(rect()); red=Real, blue=Imag')
axis([0 (length-1) -75 320])
grid on;

print -dtiff 5.tiff
```
cos/impulse Pair

```matlab
%--- rect 10 in frequency
clear;
length = 1024;
x = 0:(length-1);

f = zeros(1,length);
f(1:10) = 16;
f((length-8):length) = 16;  % "frequency" domain

fftshift of rect 10 in frequency

f = ifft(f);
figure(7);
clf;
subplot(2,1,1);
plot(x,real(t),',r');
hold on;
plot(x,imag(t),',b');
title('Time domain = IFFT(rect()); red=Real, blue=Imag');
axis([0 (length-1) -0.1 0.33]);
grid on;
subplot(2,1,2);
plot(x,f);
title('Frequency domain = rect()');
axis([0 (length-1) 0 18]);
grid on;
print -dtiff 7.tiff

figure(8);
clf;
subplot(2,1,1);
plot(x,real(t),',r');
hold on;
plot(x,imag(t),',b');
fftshift of rect 10 in frequency

plot([511 511], [1 -1], ',k--', ',linewidth', 1.5);
title('Time domain = fftshift(IFFT(rect())); red=Real, blue=Imag');
axis([0 (length-1) -0.1 0.33]);
grid on;
subplot(2,1,2);
plot(x,f);
title('Frequency domain = rect()');
axis([0 (length-1) 0 18]);
grid on;
print -dtiff 8.tiff
```