SPRING 2023

CS 378: INTRO TO SPEECH AND AUDIO PROCESSING

Digital Signal Processing for Speech 1

DAVID HARWATH
Assistant Professor, UTCS

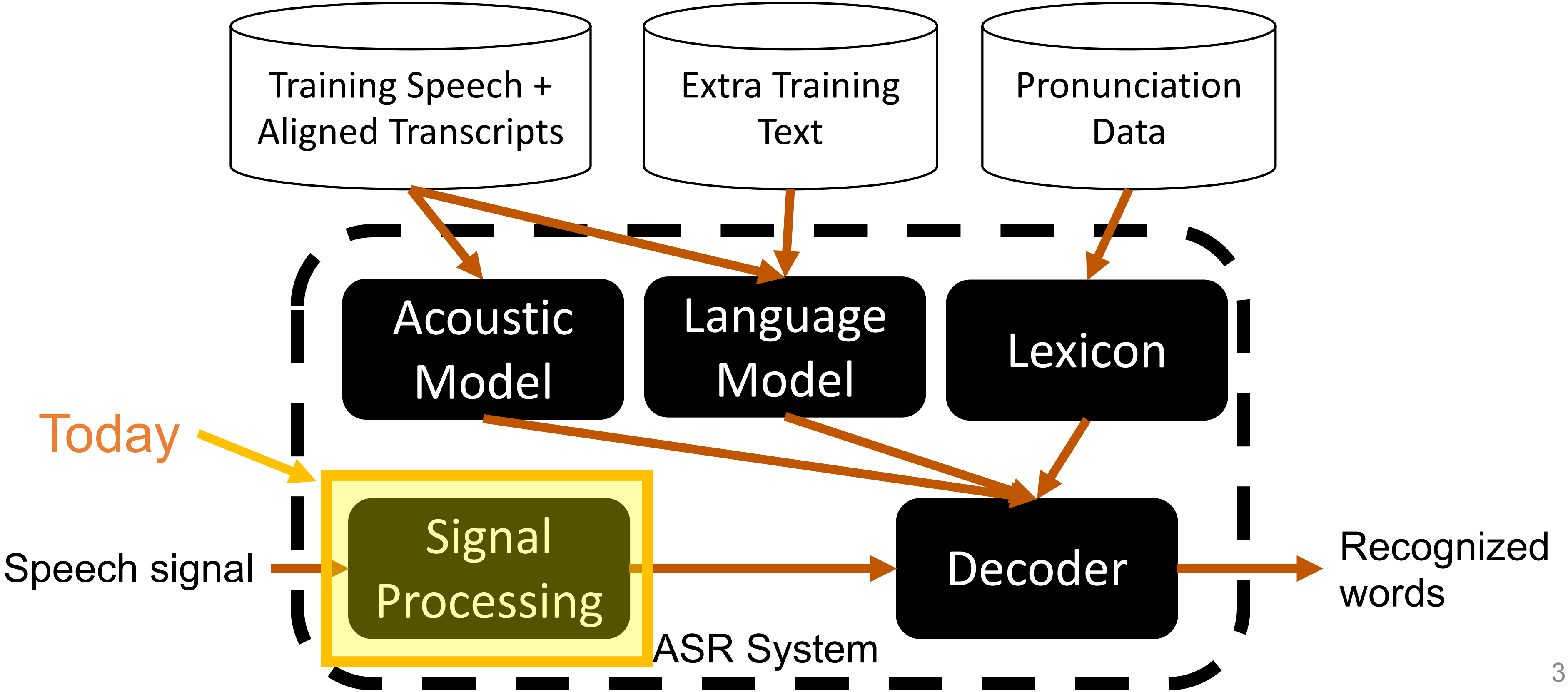
The University of Texas at Austin
Department of Computer Science
College of Natural Sciences

Today’s agenda

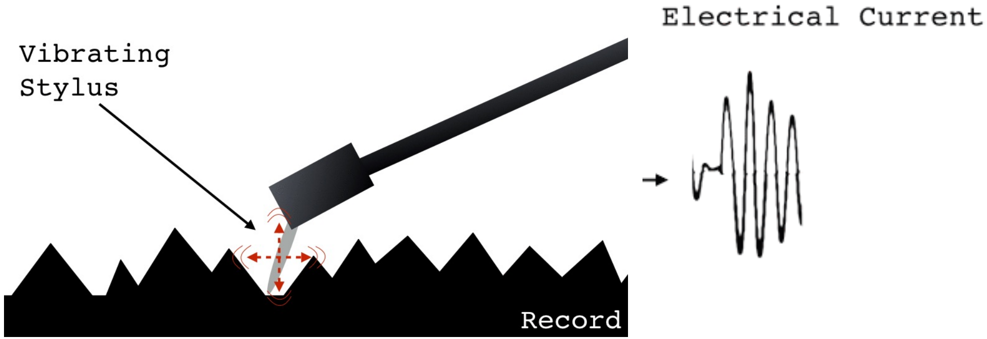
- We will cover the basics of Digital Signal Processing (DSP) that you need to understand the so-called “front-end” of a speech recognition system

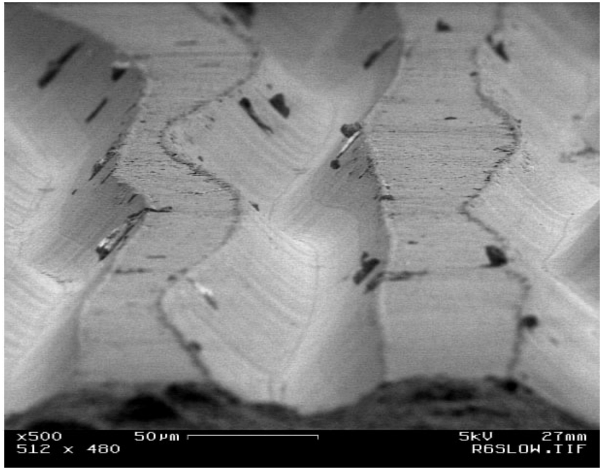
- We won’t cover DSP in nearly as much depth as a dedicated course would.
  - But we will cover some speech-specific techniques that a DSP course may not get to.

Components of an ASR system



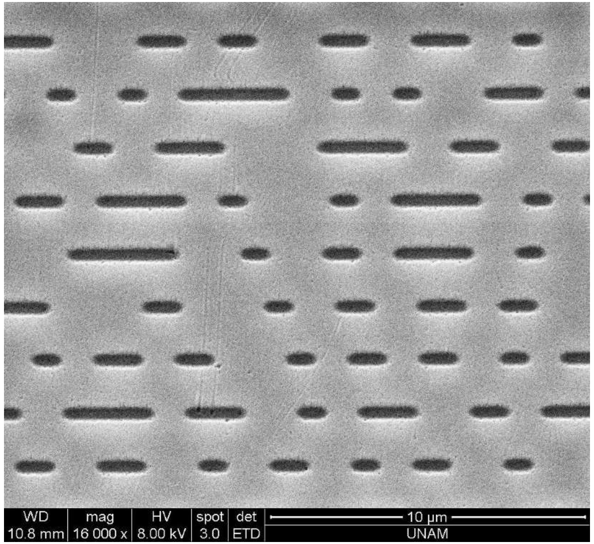
Analog vs. Digital Media





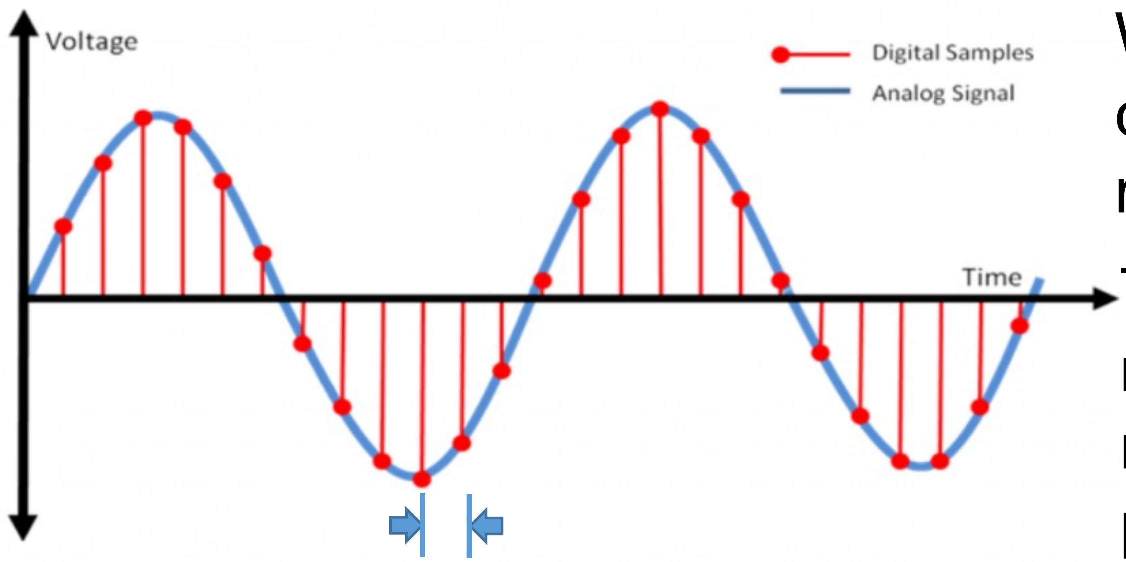
Analog media stores information *continuously* – as a pattern of physical etchings, varying magnetization of tape, etc.

Analog vs. Digital Media



Digital media stores information as a discrete sequence of binary values

Digital Representation of Audio



Sampling period $T_s = \frac{1}{f_s}$

We can use *sampling* to create a *discrete time* representation of a signal

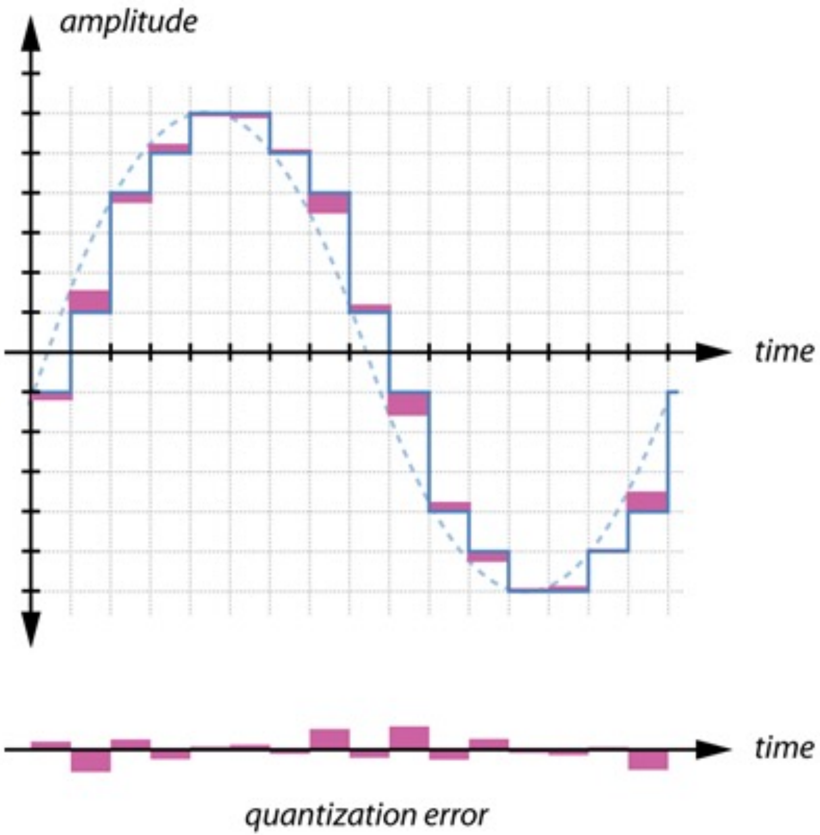
The sampling rate of a recording refers to the number of samples taken per second.

Continuous time signal $x(t)$

$\Downarrow$

Discrete time signal $x[n] = x(nT_s)$

Digital Representation of Audio



We also need to discretize the signal in *amplitude*. This process is called *quantization.*

The *bit depth* of a quantized audio signal refers to how many quantization levels are available.

16 bit depth → $2^{16} = 65536$ quantization levels

Implications of Discretization

- Discretization in amplitude introduces additive noise
  - $x_q[n] = x[n] + \epsilon_q[n]$
  - As long as we use enough quantization bits, $\epsilon_q[n]$ will be small enough that we can ignore it.
  - 16 bit depth gives approximately 96 dB SNR

- Discretization in time limits the range of frequencies that we can capture (Nyquist Sampling Theorem)
  - It also forces us to modify the Fourier Transform computation

Defining Sampling Mathematically

We have continuous time signal $x(t)$

We sample $x(t)$ by evaluating it at a series of evenly spaced intervals:

$$x[n] = x(nT_s) \;\; \text{for n} = -\infty, \dots, -2, -1, 0, 1, 2, \dots, +\infty$$

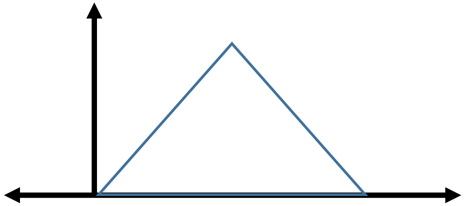
Where T_s is a constant (the *sampling period)*

This is equivalent to multiplying $\mathrm{x}(t)$ with an *impulse train* $s(t)$:

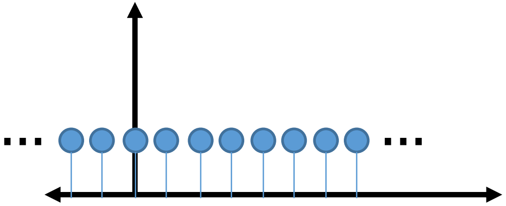
$$\mathrm{s}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

Defining Sampling Mathematically

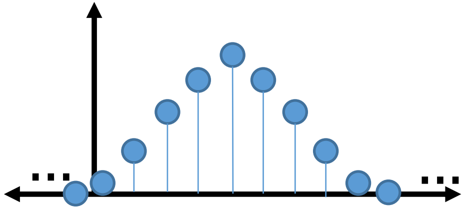
$$x[n] = x(t)s(t) = \sum_{n=-\infty}^{\infty} x(t)\delta(t - nT_s)$$



$x(t)$



$s(t)$



$x[n]$

Recall: Continuous Time Fourier Transform (CTFT)

- The Continuous Time Fourier Transform (CTFT):

$$X(\Omega) = \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt$$

- Both $x(t)$ and $X(\Omega)$ are continuous in their argument

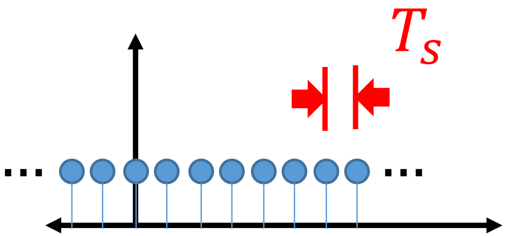
- $X(\Omega)$ is defined over $\Omega \in (-\infty, +\infty)$

The CTFT of a sampled signal

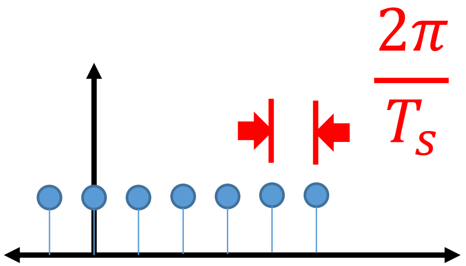
$$\mathrm{CTFT}(x[n]) = \mathrm{CTFT}(x(t)s(t))$$

Recall: Multiplication in time domain ↔ Convolution in frequency domain

$$\mathrm{CTFT}(x[n]) = \mathrm{CTFT}\big(x(t)\big) * \mathrm{CTFT}(s(t))$$

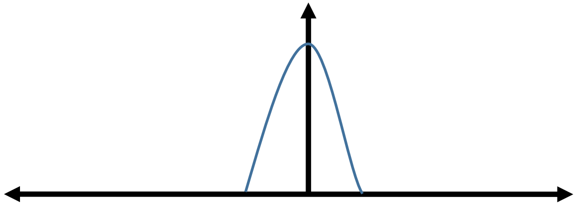


$s(t)$

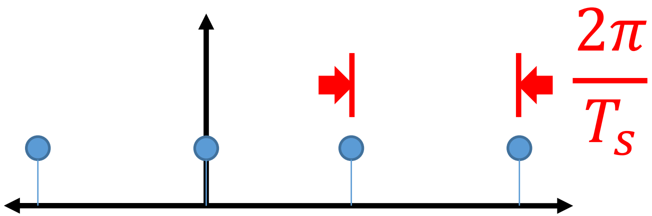


$S(\Omega)$

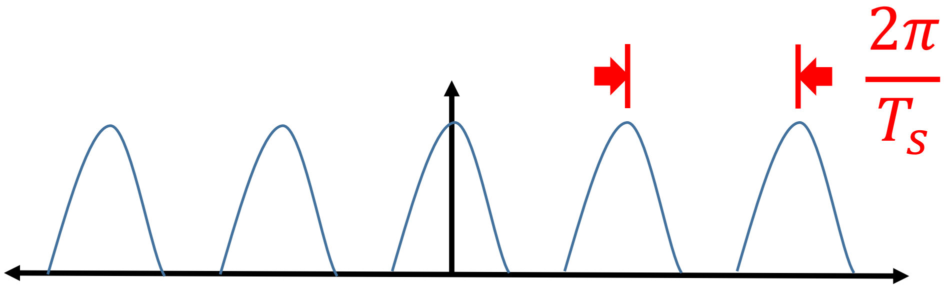
The CTFT of a sampled signal



$X(\Omega)$



$S(\Omega)$



$X(\Omega) * S(\Omega)$

Taking the CTFT of a sampled signal $x(t)s(t)$ results in taking the original spectrum $X(\Omega)$ and "copy-pasting" it at intervals $\frac{2\pi}{T_s}$

The Discrete Time Fourier Transform

Let's derive an expression for the CTFT of $x[n] = x(t)s(t)$

$$\mathrm{CTFT}(\mathrm{x}(t)s(t)) = \int_{-\infty}^{\infty} x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s)\, e^{-j\Omega t} dt$$

$$= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s)\, e^{-j\Omega t} dt$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) \int_{-\infty}^{\infty} \delta(t - nT_s) e^{-j\Omega t} dt$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s)\, e^{-j\Omega n T_s}$$

The Discrete Time Fourier Transform

We have that

$$\text{CTFT}(\text{x}(t)s(t)) = \sum_{n=-\infty}^{\infty} x(nT_s)\, e^{-j\Omega n T_s}$$

Let's substitute $\omega = \Omega T_s$ (we will call ω "digital frequency) and call this new function the "Discrete Time Fourier Transform" (DTFT)

$$\text{DTFT}(\text{x[n]}) = \text{X}(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

The Discrete Time Fourier Transform

- The Discrete Time Fourier Transform (DTFT) is defined as:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

- $x[n]$ is a discrete time signal, but $X(\omega)$ is continuous

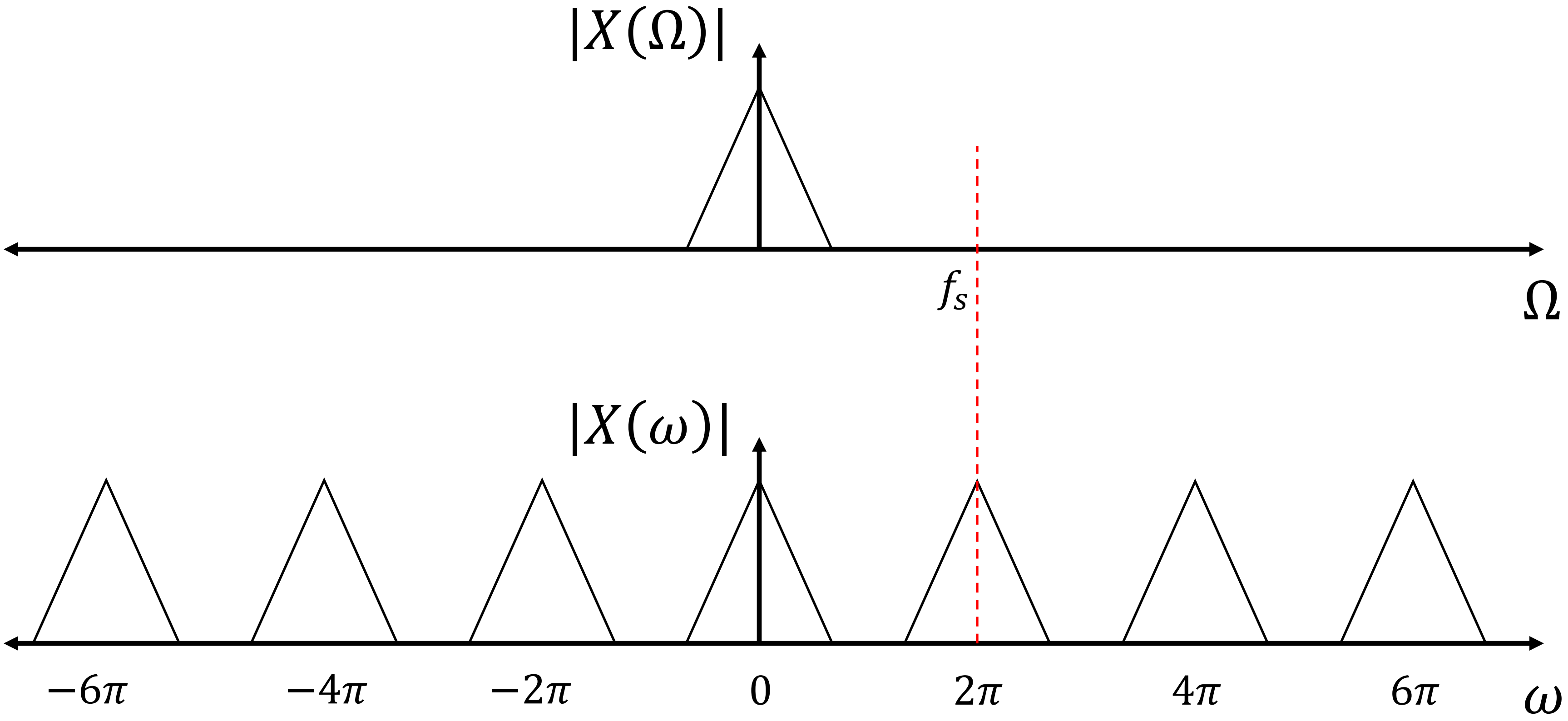
- $X(\omega)$ is defined over $\omega \in [-\infty, \infty]$, **but is periodic with period 2π, which corresponds to $\Omega = f_s$**

Inverse DTFT

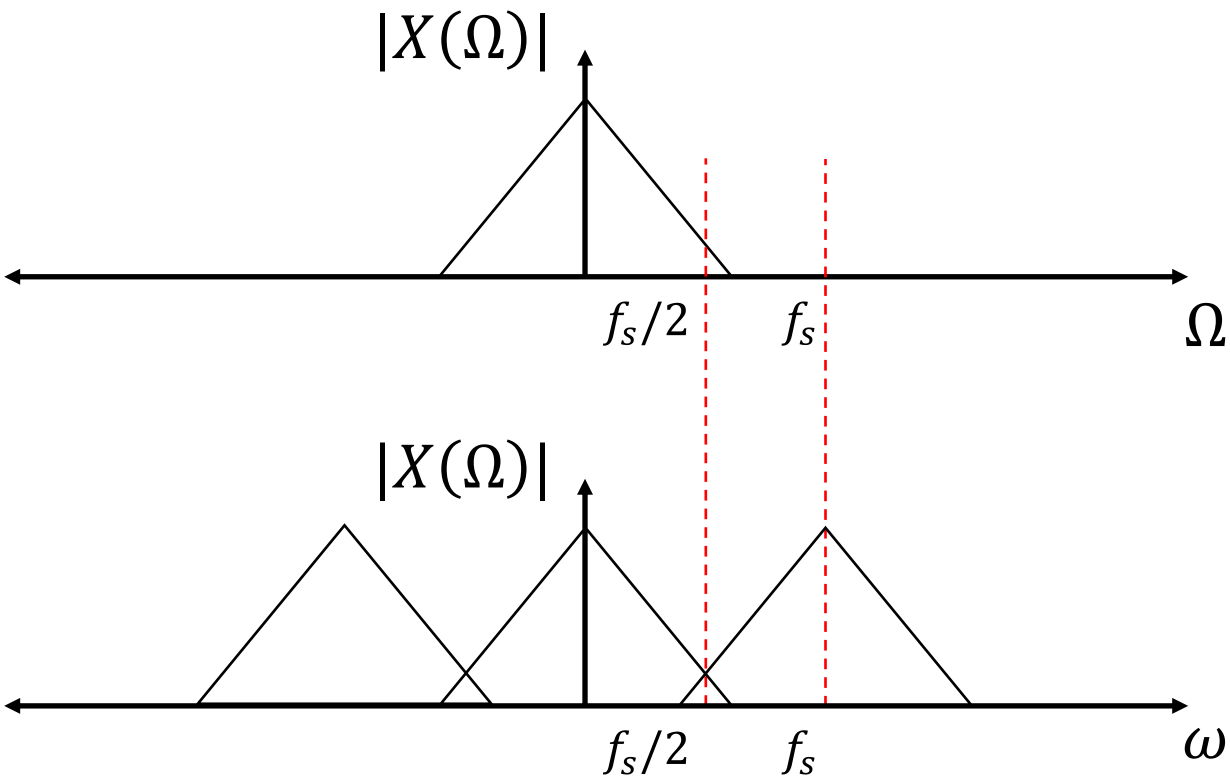
- The Discrete Time Fourier Transform is an invertible transform. We can recover $x[n]$ from $X(\omega)$ using the Inverse DTFT (IDTFT):

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}d\omega$$

Periodicity of DTFT



Aliasing

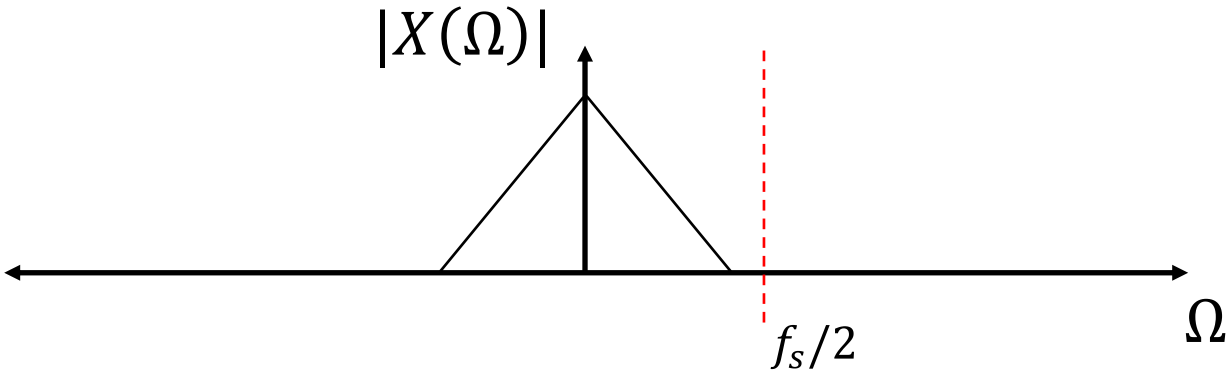


Nyquist Sampling Theorem

- The Nyquist Sampling Theorem states that we can capture **all** of the information in the continuous signal $x(t)$ with the sampled signal $x[n] = x(nT_s)$, provided that we sample fast enough

- The critical sampling rate is known as the *Nyquist rate* and is equal to **twice the highest frequency present in** $\mathrm{X}(\Omega)$

Sampling Theorem Implications



We won't run into any problems with sampling as long as we:

1. Set $f_s > 2\Omega_{max}$ where Ω_{max} is the highest frequency we want to be able to measure
2. Lowpass filter $x(t)$ to eliminate all frequencies greater than Ω_{max} before performing any sampling

DTFT Convolution Theorem

- The convolution theorem for the DTFT is slightly different, because the convolution of periodic spectra in the digital frequency domain will have infinite energy
- Instead we have a duality between multiplication in time and *periodic convolution* in frequency:

$$z[n] = x[n]y[n]$$

$$\Updownarrow$$

$$Z(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} Y(\omega)X(\omega-\theta)d\theta$$

$$z[n] = x[n] * y[n]$$

$$\Updownarrow$$

$$Z(\omega) = Y(\omega)X(\omega)$$