What is missing: abstract manipulation in code space without specifying how to map to physical signals

→ digital signals (square waves) or analog signals

→ ignore since not designing complete CDMA system

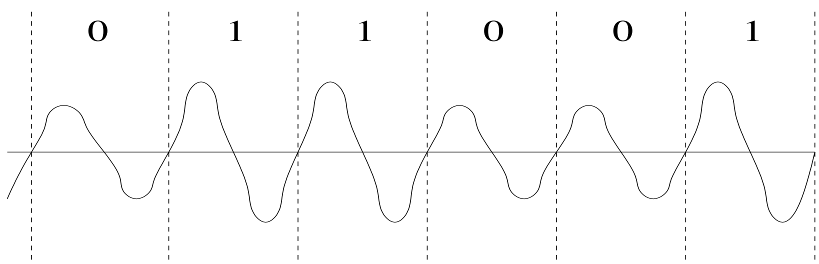
Next: borrow the conceptual framework from linear algebra for hiding bits in electromagnetic waves

→ signals: complex sinusoids

→ good news: much of the conceptual framework carries over

Back to sending bits using EM waves

→ e.g., amplitude modulation to send bits 011001



Key variable: frequency f of sinusoid

→ $\sin ft$

Important: once we change amplitude over time to carry bits, resultant signal is not a pure sine curve anymore

→ some function of time

→ signal $s(t)$

To transmit multiple bit streams concurrently:

→ use multiple frequencies $f_1, f_2, \ldots, f_n$

→ called carrier frequency

→ FDM (frequency division multiplexing)

→ called WDM (wave division multiplexing) for optical fiber

Two primary application scenarios:

- multi-user: one user gets one frequency

  → FDMA (frequency division multiple access)

- single-user: one user gets all frequencies

  → ship bits in parallel

  → completion time to ship group of bits is reduced

Other applications:

- confidentiality: protect against eavesdropping by randomly jumping around multiple frequencies

  → frequency hopping

  → transmission is sequential

- anti-jamming: spreading bits over multiple frequencies makes jamming harder
- frequency-selective fading

  → some frequencies suffer more distortion than others

  → use error correction

Referred to as spread spectrum

→ bits are spread over a wide band of frequencies

→ width from f_1 to f_n: bandwidth

→ e.g., $n = 10$, $f_1 = 1$ GHz, $f_{10} = 1.9$ GHz, bandwidth 0.9 GHz

Engineering (and fundamental) caveat: nature cannot be accurately captured by real numbers

$\rightarrow$ must use complex numbers to make sense

Use complex sinusoids

$\rightarrow \cos ft + i \sin ft$

$\rightarrow$ by Euler's formula

$$e^{ift} = \cos ft + i \sin ft$$

What is the correspondence with linear algebra used in CDMA?

In CDMA linear algebra:

- finite dimension n: number of users
- fix basis vectors $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n$: code vectors

  $\rightarrow$ orthogonal
- any vector $\mathbf{z}$ is a weighted sum of basis vectors

  $\rightarrow \mathbf{z} = \sum_{k=1}^{n} a_k \mathbf{x}^k$

  $\rightarrow$ $\mathbf{z}$ was constructed by manipulating the scalar weights a_k of $\mathbf{x}^k$ for all n to hide bits

  $\rightarrow$ $\mathbf{z}$ encodes n bits: message

In FDMA with complex sinusoids:

- infinite dimensional space

  $\rightarrow$ also continuous time (since sinusoids)

- basis elements: complex sinusoids e^{ift} for different f

  $\rightarrow$ frequency is also continuous

- many signals $s(t)$ of interest are weighted sum of e^{ift}

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} a(f) e^{ift} df$$

Note: shape is analogous to $\mathbf{z} = \sum_{k=1}^{n} a_k \mathbf{x}^k$

$\rightarrow$ weighted sum

$\rightarrow$ $\sum$ becomes $\int$ since f is continuous

$\rightarrow$ called inverse Fourier transform

$\rightarrow$ as before: hide bits in the weights $a(f)$

$\rightarrow$ sender task called synthesis

Given signal $s(t)$, our message, in whose coefficients we have hidden bits, how to recover the bits:

Perform Fourier transform:

$$a(f) = \int_{-\infty}^{\infty} s(t)e^{-ift}dt$$

$\rightarrow$ similar to inner product of $s(t)$ and e^{ift}

$\rightarrow$ $a(f) = s(t) \circ e^{ift}$

$\rightarrow$ receiver task called analysis

How to calculate Fourier transform quickly is important

$\rightarrow$ fast Fourier transform (FFT)

$\rightarrow$ same goes for synthesis: IFFT

$\rightarrow$ subject of CS580

Back to data transmission application:

$\rightarrow$ to send n bits, we need only n frequencies $f_1, f_2, \ldots, f_n$

$\rightarrow$ discrete

$$s(t) = \sum_{k=1}^{n} a_k e^{i f_k t}$$

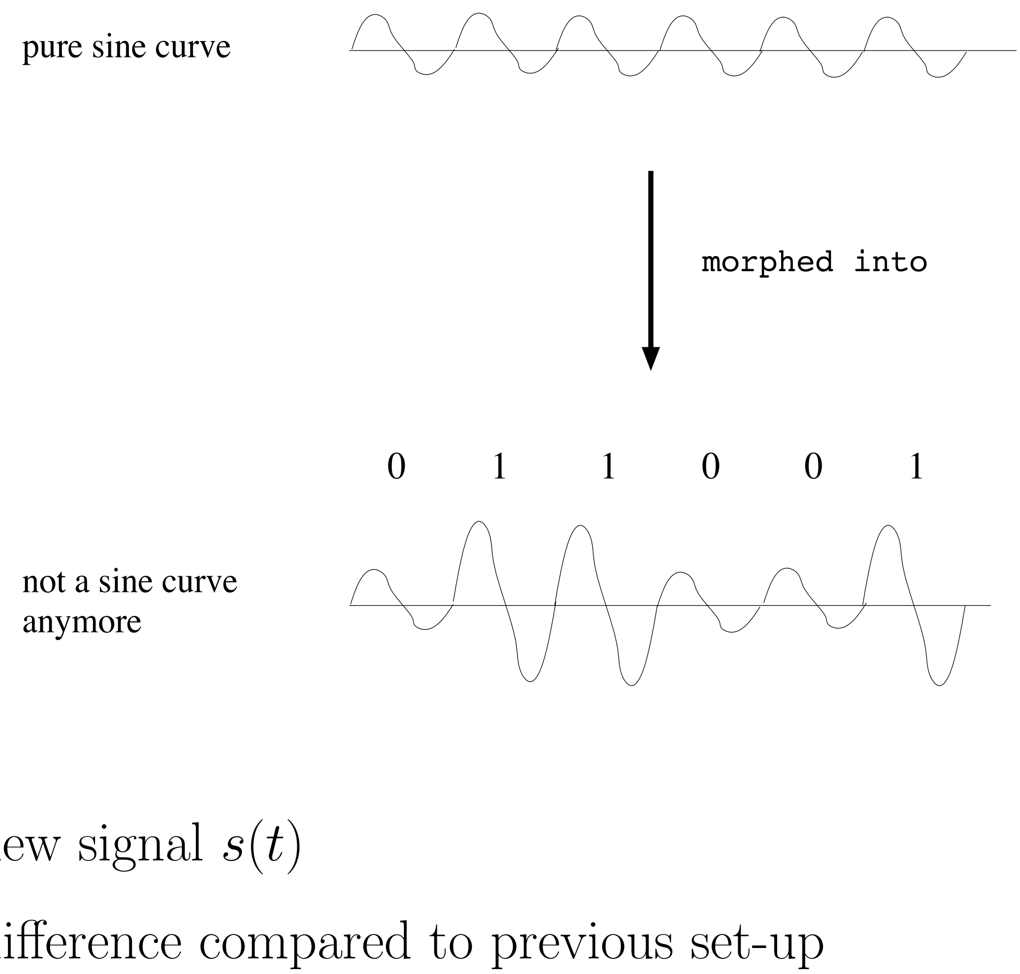
How to choose n carrier frequencies $f_1, f_2, \ldots, f_n$?

$\rightarrow$ traditional method

$\rightarrow$ modern method: OFDM (orthogonal FDM)

Traditional FDM:

Consider AM modulation of single sinusoid of frequency f, say $f = 100$ MHz

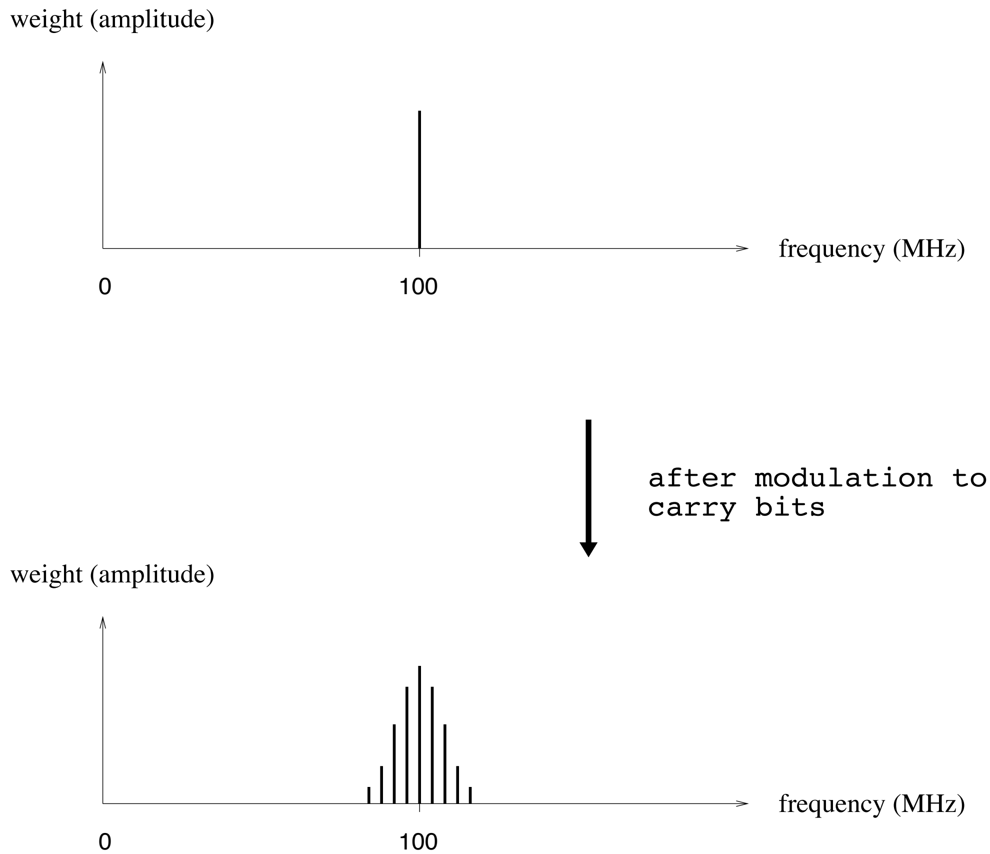


$\rightarrow$ new signal $s(t)$

$\rightarrow$ difference compared to previous set-up

$\rightarrow$ spectrum before and after AM?

Sending single bit vs. stream of bits:



Changing amplitude over time to encode bits produced non-sinusoid signal

$\rightarrow$ sum of multiple sinusoids

What other sinusoids (besides 100 MHz) need to be added together to get signal $s(t)$?

If $s(t)$ only needs a bounded range of frequencies

$\rightarrow$ bandlimited

$\rightarrow$ bounded range is called its bandwidth (Hz)

Not to be confused with

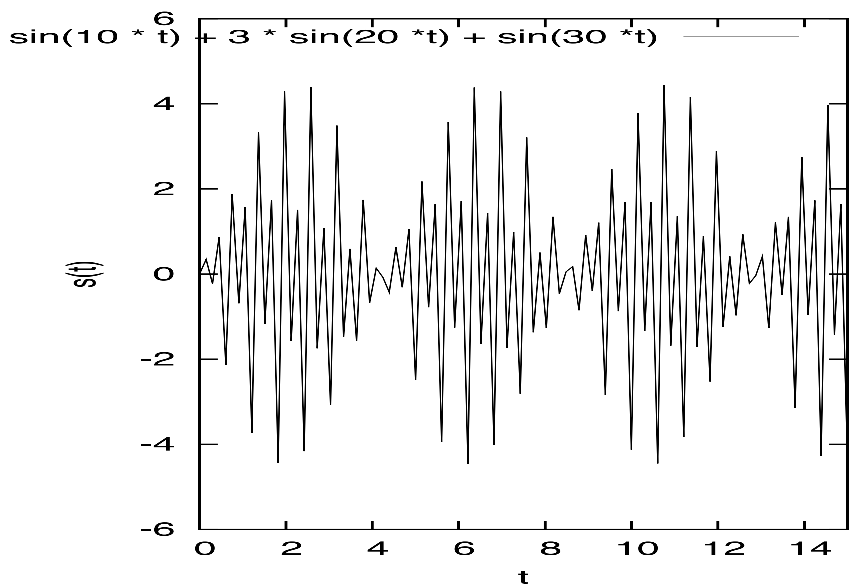
- bps bandwidth (e.g., 1 Gbps link)
- transmission medium (optical fiber, copper, wireless) characterized by bandwidth (Hz)
- fidelity of output signal to input signal

Much of communication engineering deals with bandlimited signals

$\rightarrow$ when not bandlimited approximate as bandlimited

Example: signal created by

$\rightarrow s(t) = \sin 10t + 3 \sin 20t + \sin 30t$



Spectrum of signal

$\rightarrow$ 10 Hz: 1

$\rightarrow$ 20 Hz: 3 (contributes most)

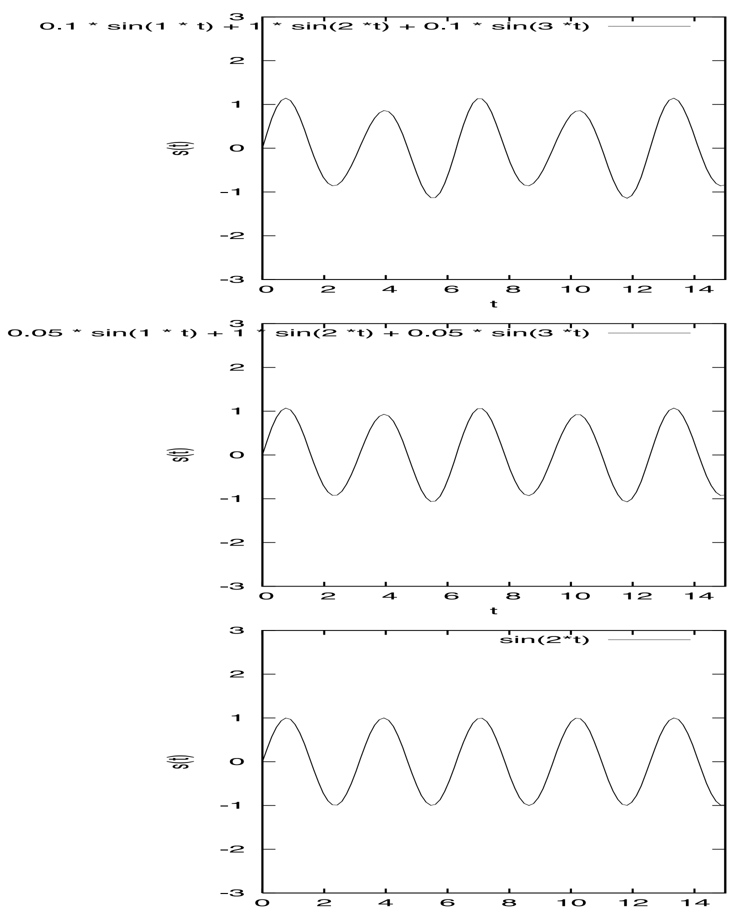
$\rightarrow$ 30 Hz: 1

Example: three signals created by

$\rightarrow s(t) = 0.1 \sin 1t + \sin 2t + 0.1 \sin 3t$

$\rightarrow s(t) = 0.05 \sin 1t + \sin 2t + 0.05 \sin 3t$

$\rightarrow s(t) = \sin 2t$

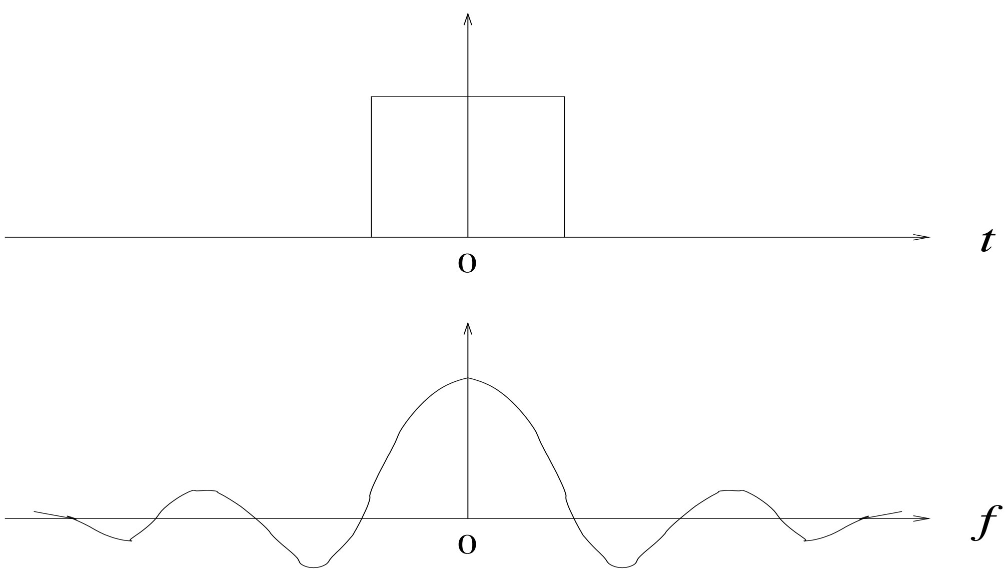


Sinusoids with small weights don't contribute much

$\rightarrow$ ignore: approximation

$\rightarrow$ treat as if weights are zero

Example: spectrum of square wave



$\rightarrow$ signal considered difficult to synthesize using sinusoids

$\rightarrow$ infinite spectrum

$\rightarrow$ cut-off and approximate