4/27/2015 -5/21/2015

Introduction to Signal Processing

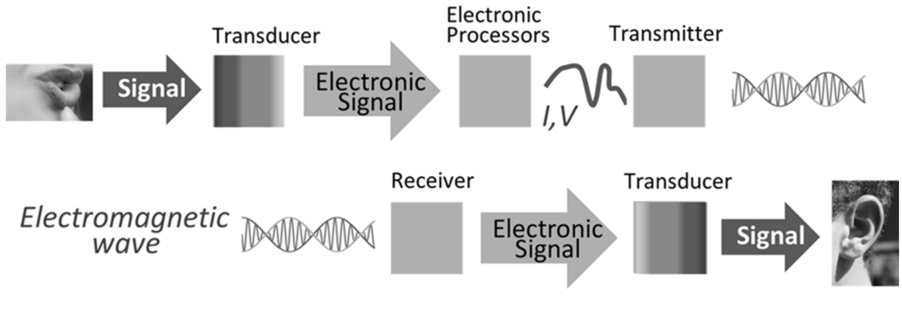
School of Information Science & Technology
ShanghaiTech University

by Xiliang Luo

信息科学与技术学院
School of Information Science and Technology

信息科学与技术学院
School of Information Science and Technology

What is Signal Processing



From Wikipedia

信息科学与技术学院
School of Information Science and Technology

Applications

❑ **Application fields of signal processing**

- **Audio/Speech signal processing**
- **Image processing**
- **Wireless communication**

- Control systems
- Array processing
- Seismology
- Financial signal processing

- **Feature extraction, Quality improvement, Compression**

信息科学与技术学院
School of Information Science and Technology

References

1. A. V. Oppenheim, A. S. Willsky and S. Hamid Nawab, Signals and Systems, Pearson Education Ltd., 2nd Edition, 2014.

2. EE16A in Berkeley: http://inst.eecs.berkeley.edu/~ee16a/sp15/

3. Single-Pixel Camera: http://dsp.rice.edu/cscamera

4. Emmanuel Candes, Compressive Sensing – A 25 Minute Tour, First EU-US Frontiers of Engineering Symposium, Cambridge, September, 2010

5. Xuedong Huang and Li Deng, An Overview of Modern Speech Recognition

6. Wikipedia on various topics: compressive sensing, speech recognition, noise cancellation

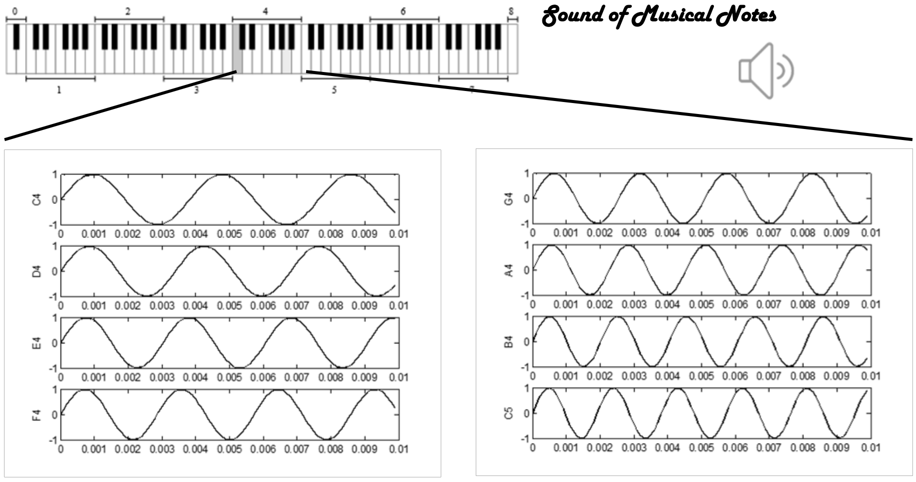
7. E. Perahia and R. Stacey, Next Generation Wireless LANs, 2nd Ed., Cambridge

Part I:
Fourier Transform & Filtering

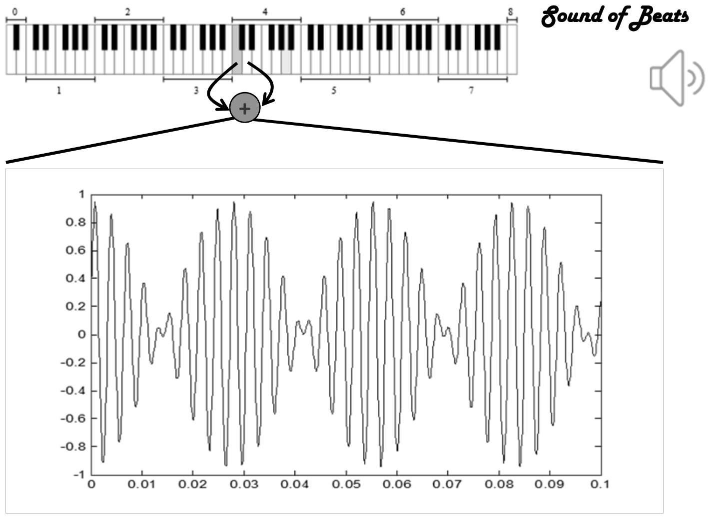
What is frequency?

How to characterize frequency?

Signals in Time Domain



Signals in Time Domain



Question:

How to define and visualize the frequency in a signal?

Answer:

Fourier Transform

/ˈfʊəriˌeɪ, -iər/

1768-1830
French Mathematician, Physicist, Historian

Frequency Domain

Fourier Transform of a signal is defined as:

Signal analysis: $$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi f t} dt$$

f: is the Frequency in Hertz = 1/second

Correspondingly, $X(f)$ can be used to recover the signal as:

Signal synthesis: $$x(t) = \int_{-\infty}^{+\infty} X(f) e^{+j2\pi f t} d\omega$$

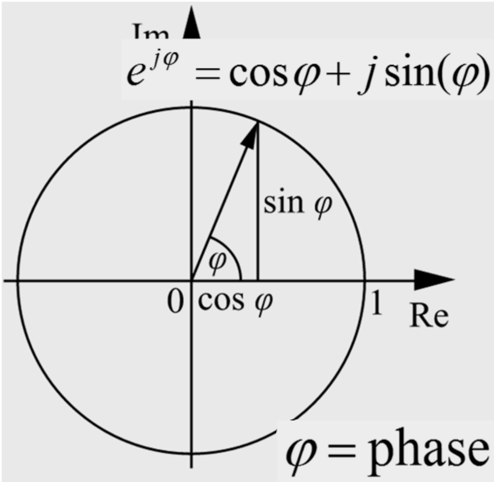
Some notations we OFTEN use

u(t) = step function = 1 only for positive t.

δ[n] = delta function (known as Kronecker Delta)
= impulse function = 1 only for n=0.

f = frequency = tells you how fast signals change
= not just how fast

Euler's Formula (frequency)



$$e^{j\varphi} = \cos\varphi + j\sin(\varphi)$$

$$\varphi = \text{phase}$$

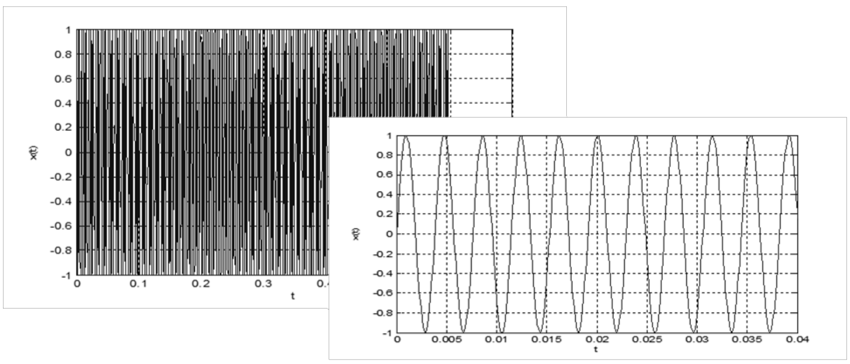
$$\cos\varphi = \frac{1}{2}(e^{j\varphi} + e^{-j\varphi})$$

$$\sin\varphi = \frac{1}{2j}(e^{j\varphi} - e^{-j\varphi})$$

Frequency Domain

Example 1: Fourier Transform of the C4 tone

$$x(t) = \begin{cases} \sin(2\pi \cdot f_0\, t), & t \in [-1,1] \\ 0, & o.w. \end{cases} \quad f_0 = 261.626\text{Hz}$$



Frequency Domain

Example 1: Fourier Transform of the C4 tone

$$X(f) = \int_{-1}^{1} \sin(2\pi f_0 t) \cdot \exp[-j2\pi f t] dt$$

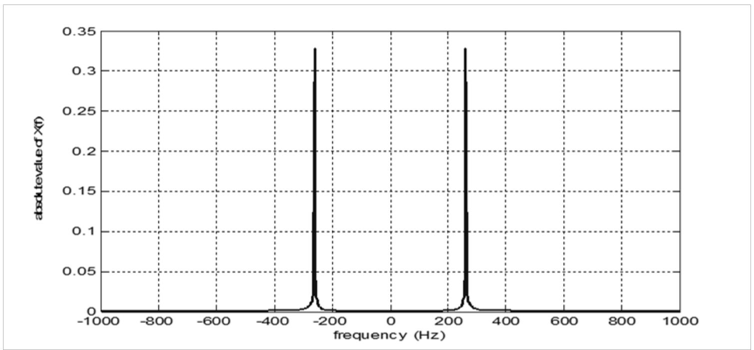
$$= \frac{1}{2j} \begin{bmatrix} \int_{-1}^{+1} \exp[j2\pi(f_0 - f)t]\, dt \; - \\ \int_{-1}^{+1} \exp[j2\pi(-f_0 - f)t]\, dt \end{bmatrix}$$

$$= \frac{\sin(2\pi(f - f_0))}{j2\pi(f - f_0)} - \frac{\sin(2\pi(f + f_0))}{j2\pi(f + f_0)}$$

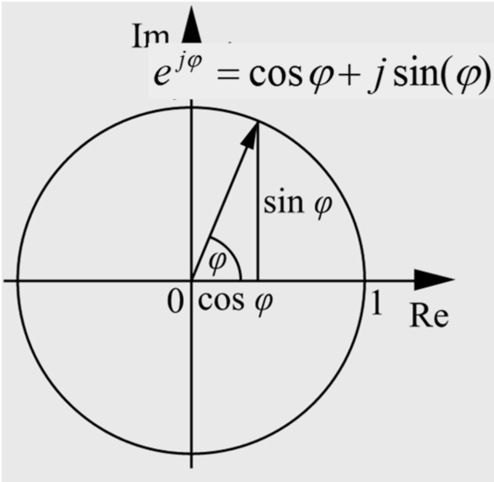
Frequency Domain

Example 1: Fourier Transform of the C4 tone

$$x(t) = \begin{cases} \sin(2\pi \cdot f_0\, t), & t \in [0,1] \\ 0, & o.w. \end{cases} \qquad f_0 = 261.626\text{Hz}$$



Euler's Formula (frequency)



$$e^{j\varphi} = \cos\varphi + j\sin(\varphi)$$

$$\cos\varphi = \frac{1}{2}(e^{j\varphi} + e^{-j\varphi})$$

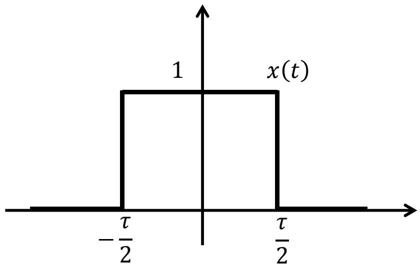
$$\sin\varphi = \frac{1}{2j}(e^{j\varphi} - e^{-j\varphi})$$

$$\varphi = 2\pi f_0 \bullet t = \text{linear phase change in } t$$

Frequency Domain

Example 2: Fourier Transform of the rectangular pulse

$$x(t) = \begin{cases} 1, & t \in \left[-\frac{\tau}{2}, \frac{\tau}{2}\right] \\ 0, & o.w. \end{cases}$$



Frequency Domain

Example 2: Fourier Transform of the rectangular pulse

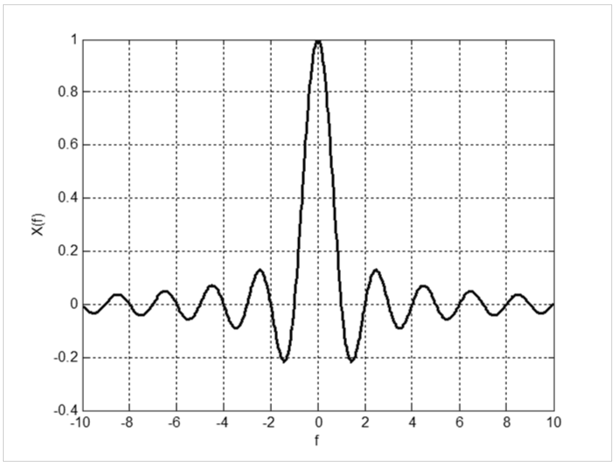
$$X(f) = \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} 1 \cdot \exp[-j2\pi f t]dt$$

$$= \frac{1}{-j2\pi f}\left[\exp\left(\frac{-j2\pi f\tau}{2}\right) - \exp\left(\frac{+j2\pi f\tau}{2}\right)\right]$$

$$= \frac{\sin(\pi f \tau)}{\pi f}$$

Frequency Domain

Example 2: Fourier Transform of the rectangular pulse

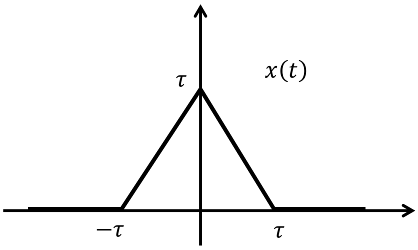


$\tau = 1$

Frequency Domain

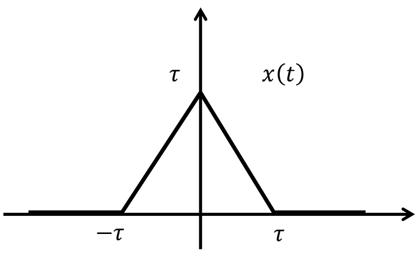
Example 3: Fourier Transform of the triangular pulse

$$x(t) = \begin{cases} \tau - |t|, & t \in [-\tau, \tau] \\ 0, & o.w. \end{cases}$$



Frequency Domain

Example 3: Fourier Transform of the triangular pulse



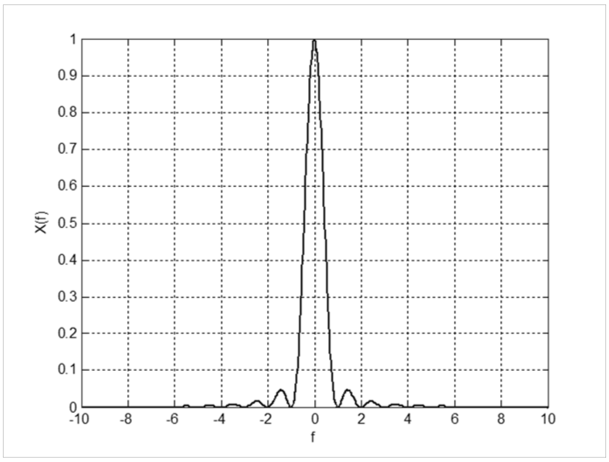
$$X(\omega) = \int_{-\infty}^{+\infty} x(t) \cdot \exp[-j2\pi f t] dt$$

$$= 2\int_0^\tau (\tau - t)\cos(2\pi f t)\, dt$$

$$= \left(\frac{\sin(\pi f \tau)}{\pi f}\right)^2$$

Frequency Domain

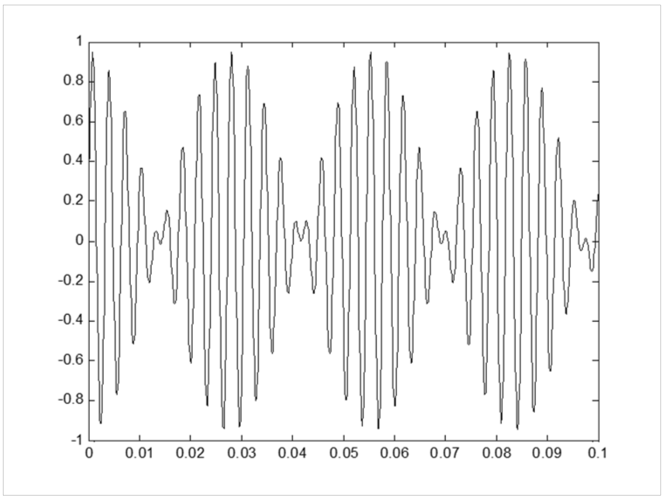
Example 3: Fourier Transform of the triangular pulse



$\tau = 1$

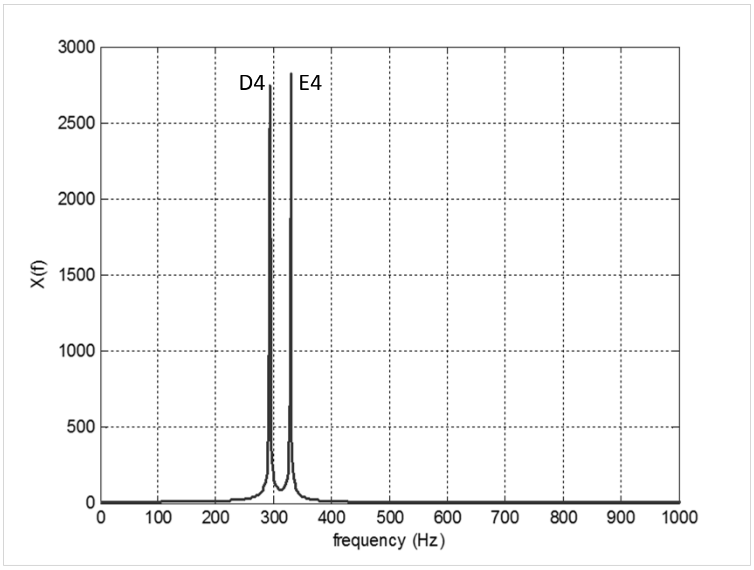
Time Domain

Back to where we begin …



Frequency Domain

Back to where we begin …



Define discrete time signals

x(t) is continuous time signal;

x[n] = x(nT) discrete signal sampled periodically

If your T is small enough, no information loss:
Nyquist sampling theorem says T < 1/(2W).

W = Bandwidth (in frequency) of a signal

Digital signal processing only deal with discrete signals

Some notations we OFTEN use

u(t) = step function = 1 only for positive t.

δ[n] = delta function (known as Kronecker Delta)
= impulse function = 1 only for n=0.

W = Bandwidth of x(t)
= maximum positive frequency for which X(f) is not zero.

Digital signal processing only deal with discrete signals

Discrete Fourier Transform

In computer, we have to discretize the signal and the following continuous Fourier Transform:

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi f t} dt$$

Sampling every T_s (F_s=1/T_s) seconds:

$$X(k) = \sum_{n=-\infty}^{+\infty} x(nT_s) e^{-j2\pi\left(\frac{kF_s}{N}\right)(nT_s)}$$
$$= \sum_{n=-\infty}^{+\infty} x(nT_s) e^{-j2\pi\left(\frac{kn}{N}\right)}$$

Fast Fourier Transform (FFT)

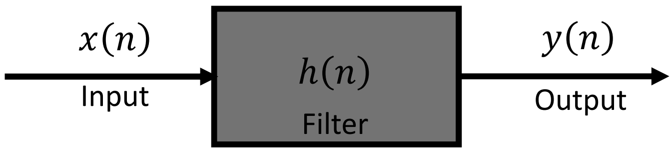
Discrete Fourier Transform:

$$X(k) = \sum_{n=-\infty}^{+\infty} x(nT_s)e^{-j2\pi\left(\frac{kn}{N}\right)}$$

- N is the power of 2 → very efficient algorithms
  - DFT is also called FFT!

Note: You will learn more during the practice session!

Filtering

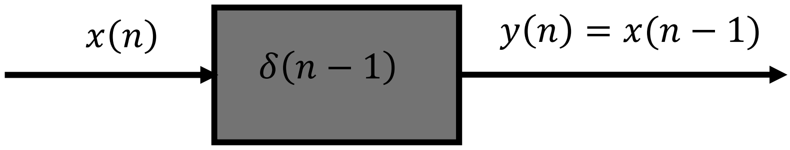


The role of a filter is to perform the following operation called "Convolution":

$$y(n) = \sum_{k=-\infty}^{+\infty} h(k)x(n-k)$$

Filtering

Example 1: Unit Delay

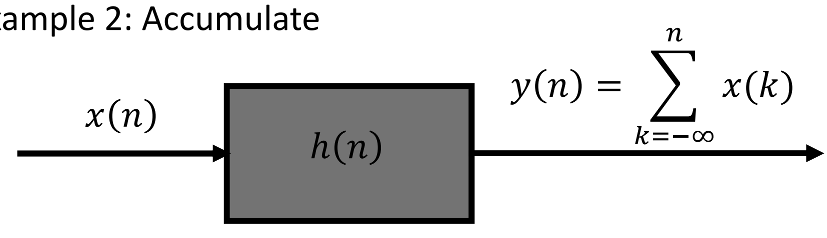


$$\delta(n-1) = \begin{cases} 1, & n = 1 \\ 0, & n \neq 1 \end{cases}$$

$$y(n) = \sum_{k=-\infty}^{+\infty} \delta(k-1)x(n-k) = x(n-1)$$

Filtering

Example 2: Accumulate

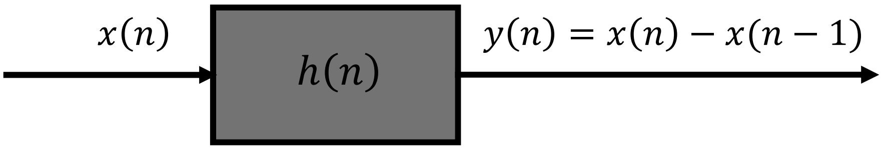


$$h(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

$$y(n) = \sum_{k=-\infty}^{+\infty} h(k)x(n-k) = \sum_{k=0}^{+\infty} x(n-k)$$

Filtering

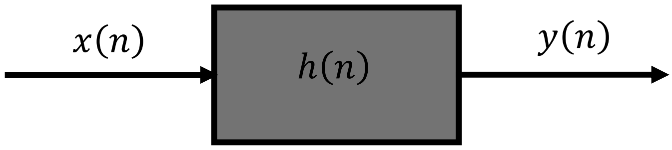
Example 3: Difference



$$h(n) = \begin{cases} 1, & n = 0 \\ -1, & n = 1 \\ 0, & o.w. \end{cases}$$

$$y(n) = \sum_{k=-\infty}^{+\infty} h(k)x(n-k) = x(n) - x(n-1)$$

Filtering Changes Fourier Transform



Time Domain	Frequency Domain
$y(n) = \sum_{k=-\infty}^{+\infty} h(k)x(n-k)$	$Y(k) = H(k)X(k)$

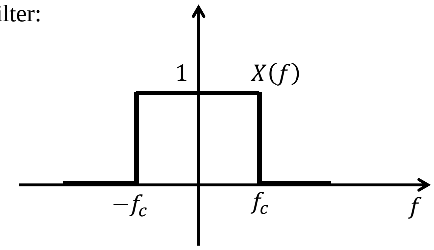
Note: You will learn more during the practice session!

Part II: Homework

信息科学与技术学院
School of Information Science and Technology

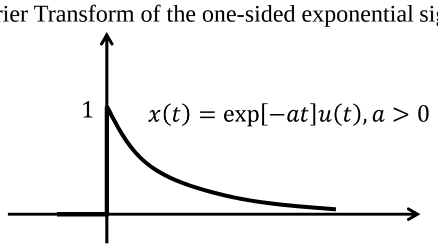
HW Problems

1. Find the time domain signal that has the following Fourier Transform, which is called Low-Pass Filter:



$$x(t) = \int_{-\infty}^{+\infty} X(f) e^{+j2\pi f t} df$$

2. Find the Fourier Transform of the one-sided exponential signal.



HW Problems

3. Find the Discrete Fourier Transform of the following signals:

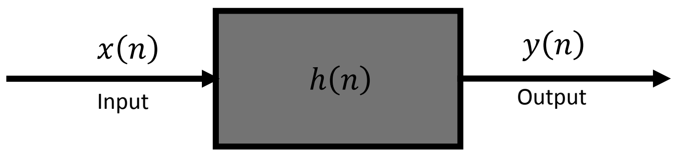
3.a $x(n) = \delta(n-1), n = 0,1,\dots,N-1$

3.b $x(n) = a^n, n = 0,1,\dots,N-1$

3.c $x(n) = \sin(2\pi f_0 n), n = 0,1,\dots,N-1$

HW Problems

4. Let y(n) be the filtering output when inputting x(n), determine the output with the following inputs:



4.a $x(n-n_0)$, n_0 is a fixed natural number

4.b $ax(n)$, a is a fixed real number

4.c $x(n) = e^{j2\pi f_0 n}$, f_0 is a fixed real number

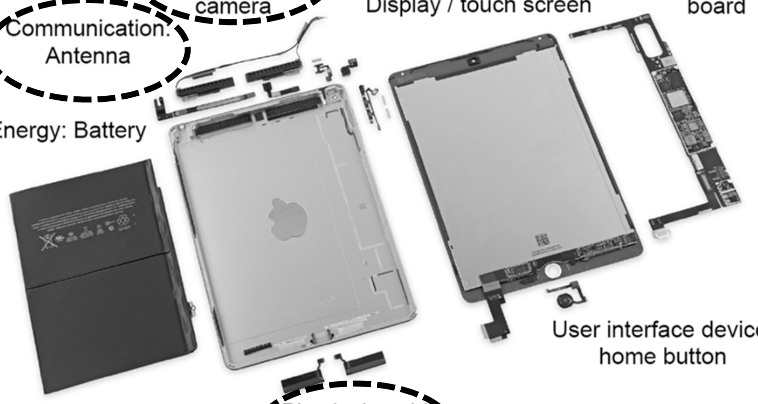
Part III: Overview of Signal Processing Applications

信息科学与技术学院
School of Information Science and Technology

Example System: IPAD Air 2

- Great stuff running apps, but:
  - What makes the display tick?
  - How does the Wi-Fi work?
  - How does it sense touch on the screen?
  - How does it sense the motion?
  - How does Siri work?
  - ...

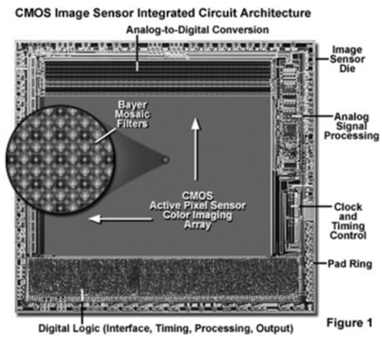
Inside an IPAD Air 2

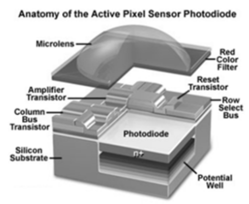


信息科学与技术学院
School of Information Science and Technology

The Camera

Goal: Convert light into electrical signals

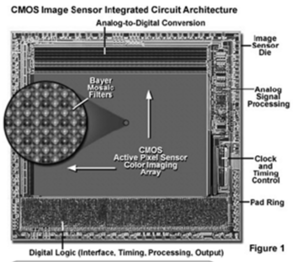




Get color spatial distribution by using an array of "light" detectors, each under a color filter

Each pixel takes the energy of incoming photons and produces charge, which is then read out by scanning through the rows and columns of the array

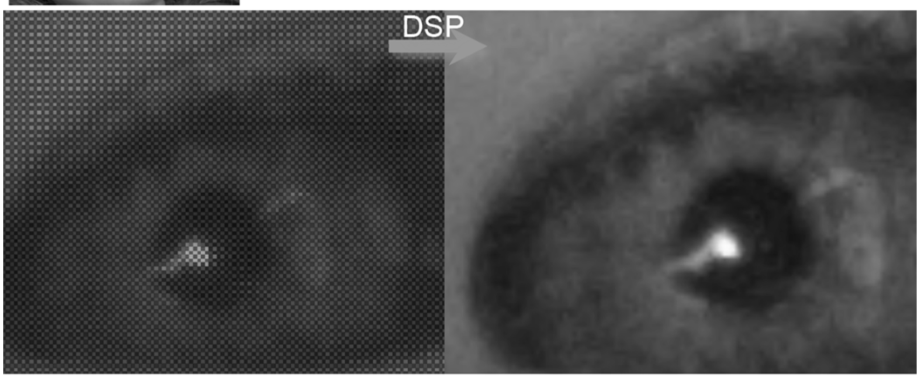
The Camera: Signal Processing



Focus/exposure Control → preprocessing → white-balancing → demosaic → Color transform → Post-processing → Compression

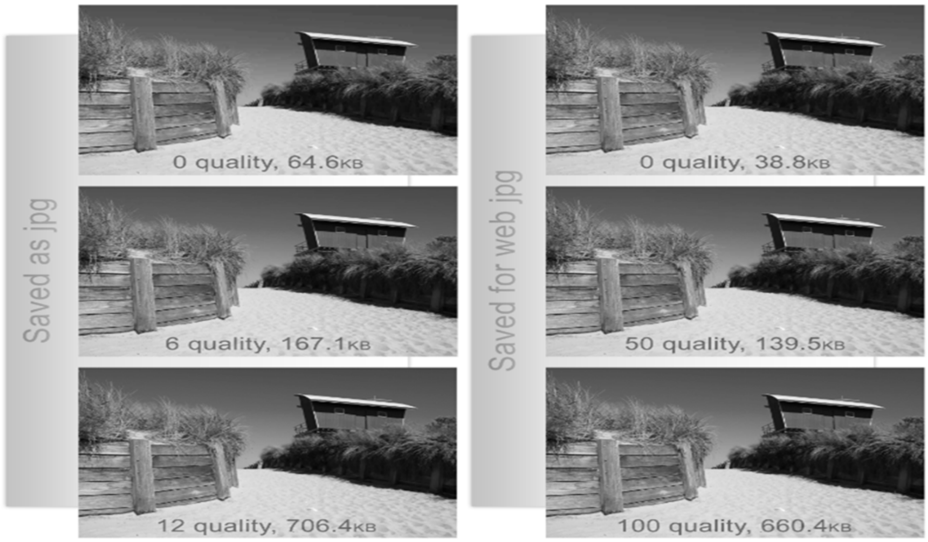
信息科学与技术学院
School of Information Science and Technology

The Camera: Power of SP



信息科学与技术学院
School of Information Science and Technology

The Camera: SP for Compression



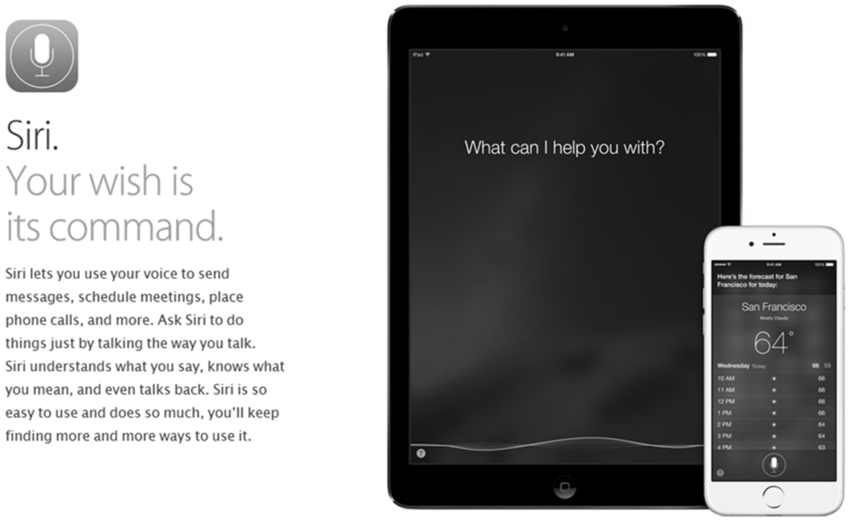
信息科学与技术学院
School of Information Science and Technology

The Camera: Comp. Photography

High Dynamic Range(HDR)

信息科学与技术学院
School of Information Science and Technology

The Microphone: Siri



信息科学与技术学院
School of Information Science and Technology

Speech Recognition

- From the technology perspective, speech recognition has been going through several waves of major innovations since over some 50 years ago.

- The most recent wave of innovations since 2009 is based on **deep learning** concepts, architectures, methodologies, algorithms, and practical system implementations enabled by big training data and by GPU-based big compute

    - defines the current state of the art in speech recognition accuracy and has been in dominant use (e.g. Siri) since 2013 throughout the speech industry worldwide

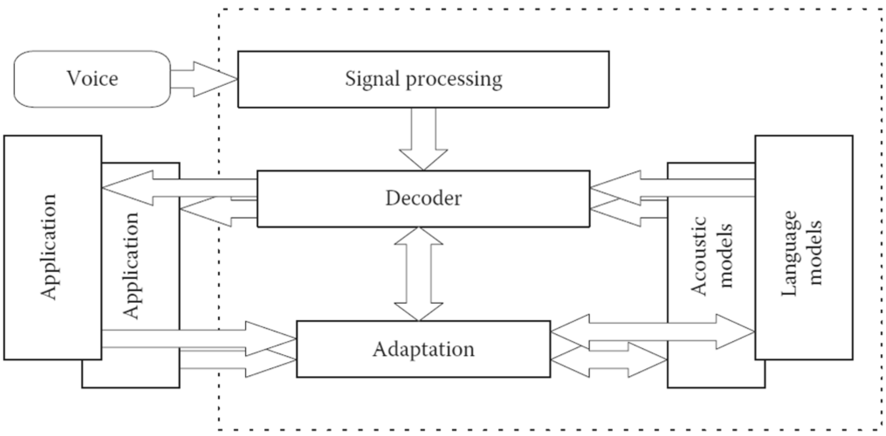
信息科学与技术学院
School of Information Science and Technology

Speech Recognition

- Applications
  - Dictation
  - Telephone-based Information (directions, air travel, banking, etc)
  - Hands-free (in car)
  - Second language ('L2') (accent reduction)
  - Audio archive searching
  - Linguistic research
    - Automatically computing word durations, etc

信息科学与技术学院
School of Information Science and Technology

Speech Recognition



Basic System Architecture of a Speech-Recognition System

信息科学与技术学院
School of Information Science and Technology

Speech Recognition

- **Fundamental of statistical speech recognition:**

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{A}) = \arg\max_{\mathbf{W}} \frac{P(\mathbf{W})P(\mathbf{A}|\mathbf{W})}{P(\mathbf{A})}$$

A: acoustic observation

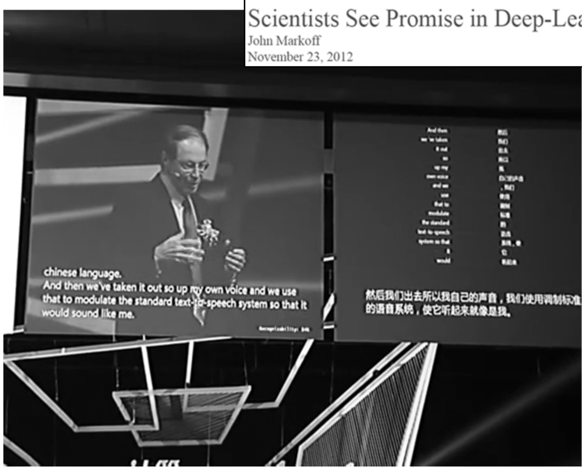
W: word sequence

The goal of speech recognition is to find out the word sequence that has the max posterior probability.

信息科学与技术学院
School of Information Science and Technology

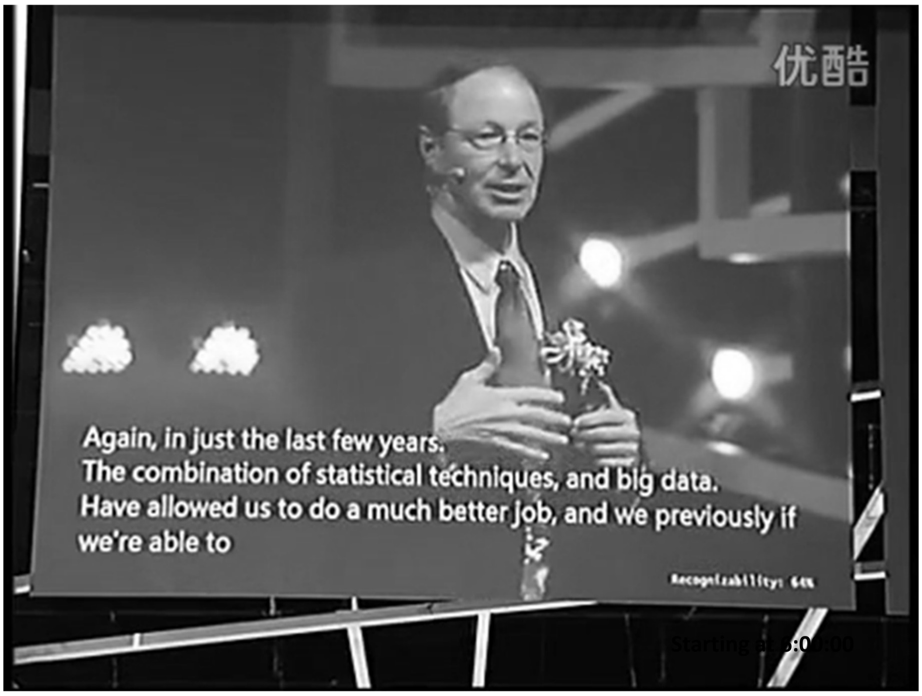
Speech Recognition

The New York Times
Scientists See Promise in Deep-Learning Programs
John Markoff
November 23, 2012



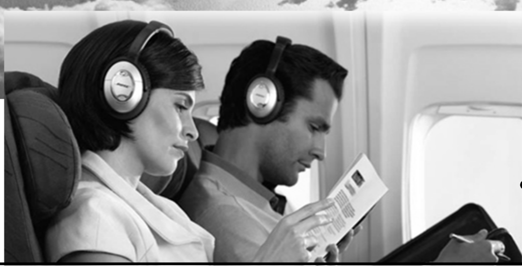
Richard F. Rashid
SVP, Microsoft
A voice recognition program translated the speech into Mandarin Chinese
Tianjin, Oct. 25, 2012

<u>Deep learning</u> technology enabled speech-to-speech translation



信息科学与技术学院
School of Information Science and Technology

The Headphone: ANC



Noise-Cancelling Headphone

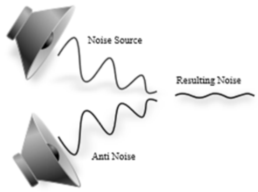
Before After

信息科学与技术学院
School of Information Science and Technology

The Headphone: ANC

Bose QuietComfort 25 review:

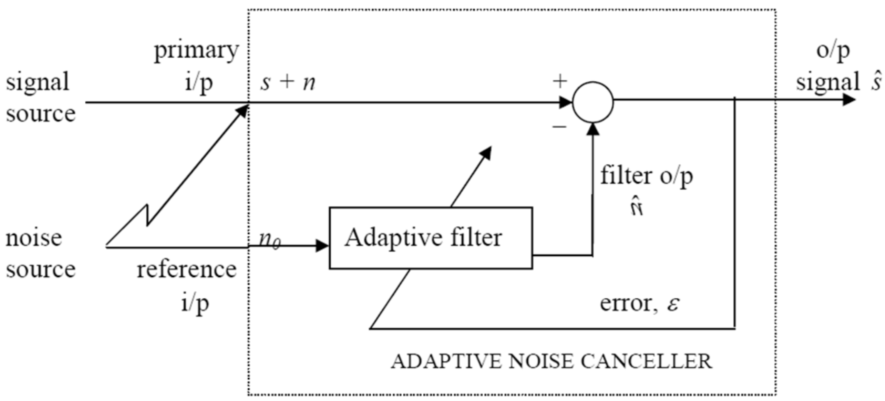
The best noise-canceling headphones get better



信息科学与技术学院
School of Information Science and Technology

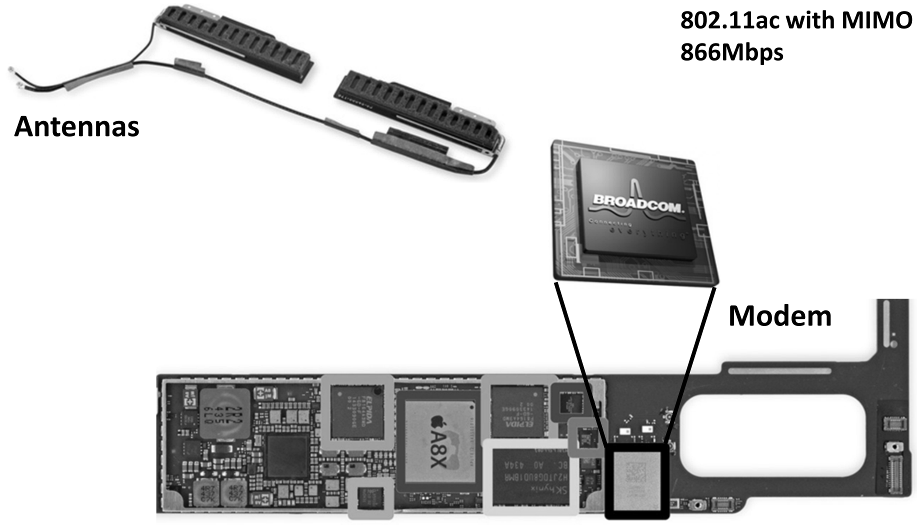
The Headphone: ANC

Adaptive Noise Canceller



信息科学与技术学院
School of Information Science and Technology

The WiFi Module



信息科学与技术学院
School of Information Science and Technology

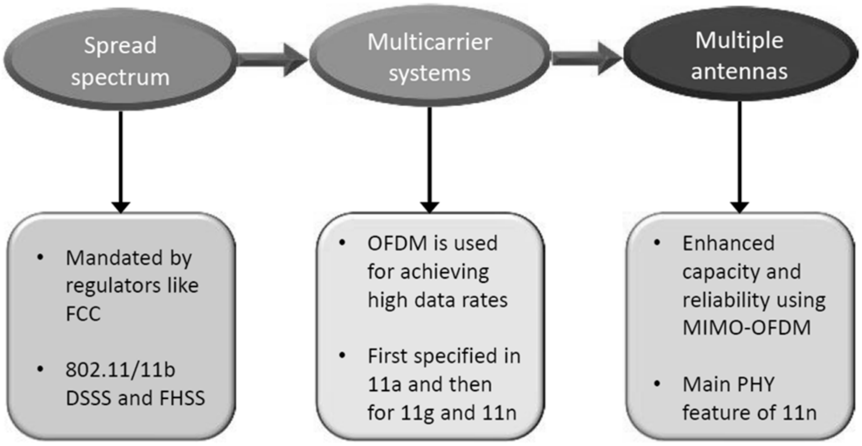
The WiFi Module

- **A brief WiFi History**
  - 1997: 802.11 standard
    - IR
      clause 15
    - FHSS,
      clause 14
    - DSSS in 2.4GHz
      2Mbps
      clause 16
  - 1999: 802.11b
    - DSSS in 2.4GHz
      11Mbps, clase 17
  - 1999: 802.11a
    - OFDM in 5GHz
      54Mbps, clase 18
  - 2003: 802.11g
    - OFDM in 2.4GHz
      54Mbps
      clase 19
  - 2009: 802.11n
    - OFDM+MIMO SM
      20MHz BW → 300Mbps
      40MHz BW → 600Mbps
      clause 20 (HT PHY) in the 2012 revision of IEEE 802.11 standard
  - 2007~now: 802.11ac
    - VHT
    - enhancing 802.11n in 5GHz

信息科学与技术学院
School of Information Science and Technology

The WiFi Module

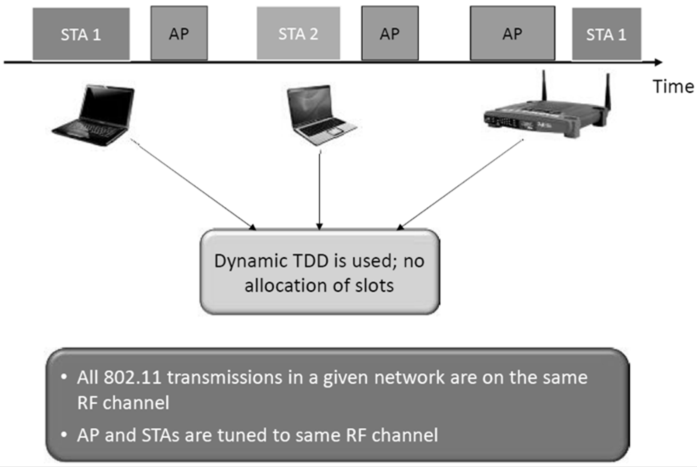
❑ A brief WiFi History



信息科学与技术学院
School of Information Science and Technology

The WiFi Module

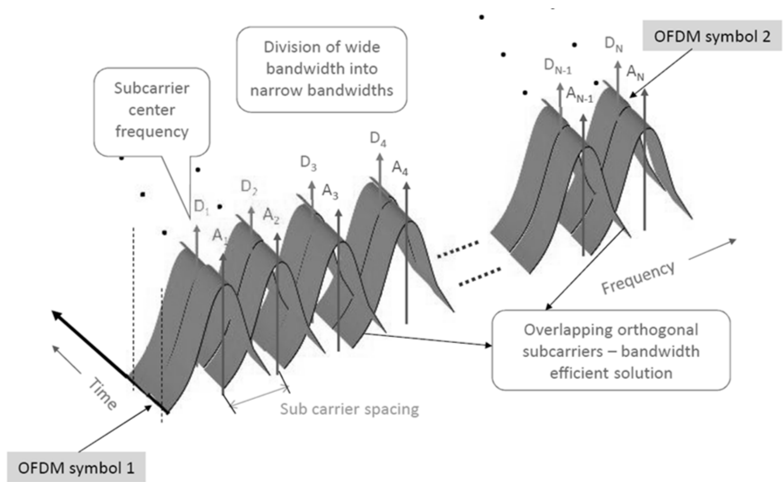
❑ Time-Division Duplexing



信息科学与技术学院
School of Information Science and Technology

The WiFi Module

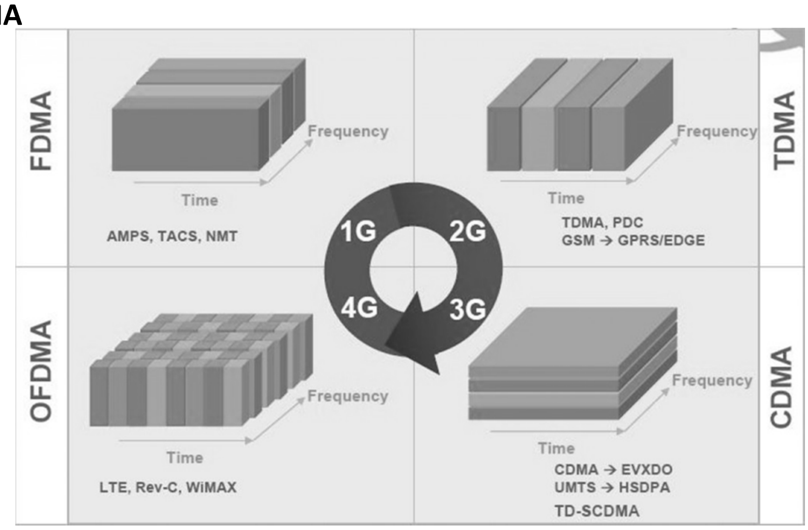
- OFDM



信息科学与技术学院
School of Information Science and Technology

The WiFi Module

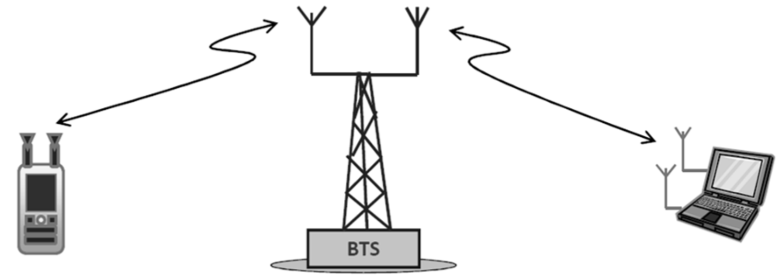
- OFDMA



信息科学与技术学院
School of Information Science and Technology

The WiFi Module

- **MIMO**



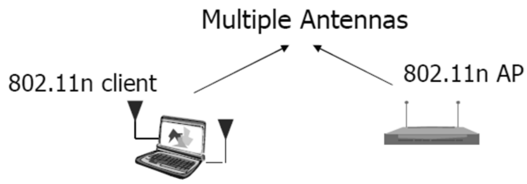
Smart signal processing/coding used at the TX./and or Rx.

Antennas are typically ordinary but DSP algorithms are important

信息科学与技术学院
School of Information Science and Technology

The WiFi Module

- **MIMO**



Using multiple antennas for multiplexing or diversity

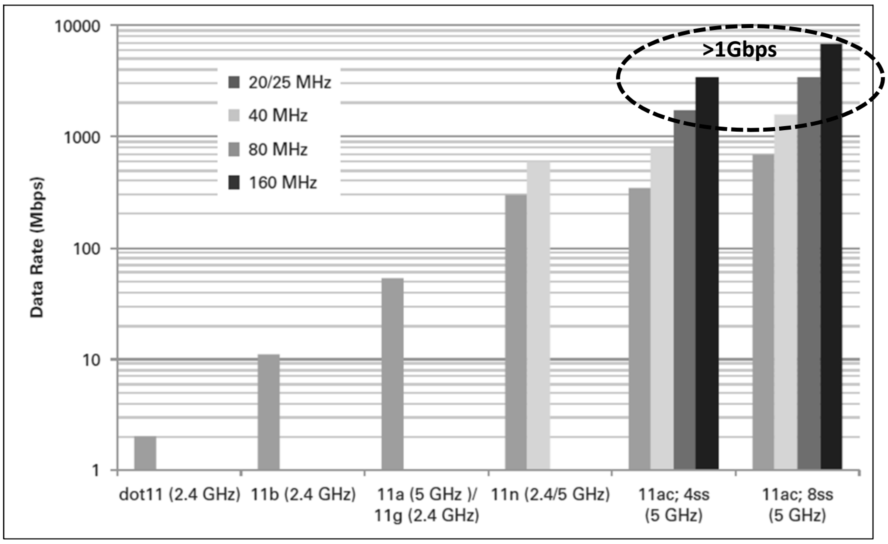
Multiplexing ⇒ Higher rates
Highly correlated channels leads to poor performance

Diversity ⇒ Better performance/range
Indirectly lead to better rates

Combining spatial multiplexing and STBC can give advantages from both

信息科学与技术学院
School of Information Science and Technology

Increase in WiFi PHY Rates

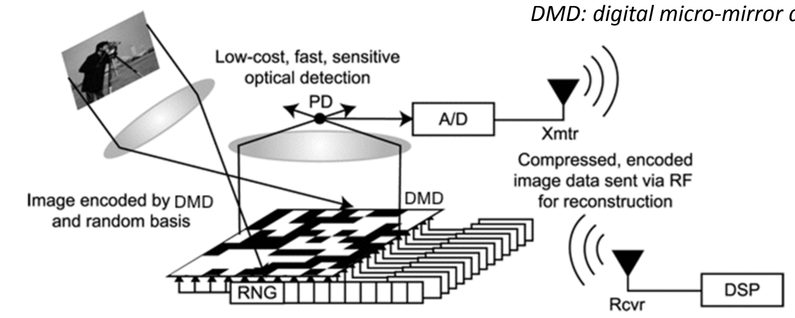


Part VI: Advanced Reading

信息科学与技术学院
School of Information Science and Technology

Single-Pixel Camera

DMD: digital micro-mirror device

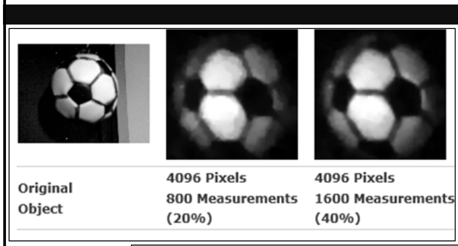


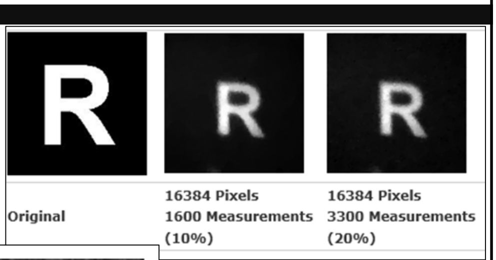
Camera directly acquires random projections of a scene without first collecting the pixels/voxels. The camera architecture employs a digital micromirror array to optically calculate linear projections of the scene onto pseudorandom binary patterns. Its key hallmark is its ability to obtain an image or video with a single detection element (the "single pixel") while measuring the scene fewer times than the number of pixels/voxels.

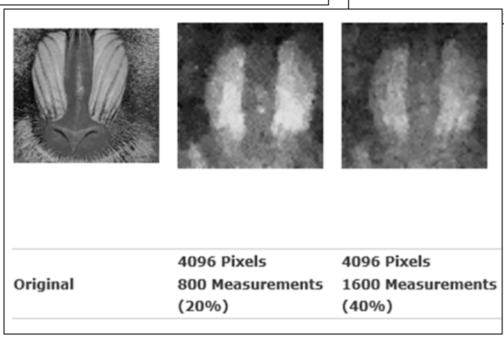
** Richard Baraniuk, Kevin Kelly, et al, http://dsp.rice.edu/cscamera*

信息科学与技术学院
School of Information Science and Technology

Single-Pixel Camera







Compressive Sensing plays the trick!

信息科学与技术学院
School of Information Science and Technology

Compressive Sensing

Compressed sensing (also known as **compressive sensing**, **compressive sampling**, or **sparse sampling**) is a signal processing technique for efficiently acquiring and reconstructing a signal, by finding solutions to underdetermined linear systems

In a nutshell …

- ❑ Can obtain super-resolved signals from just a few sensors
- ❑ Sensing is nonadaptive: no effort to understand the signal
- ❑ Simple acquisition process followed by numerical optimization

First papers:

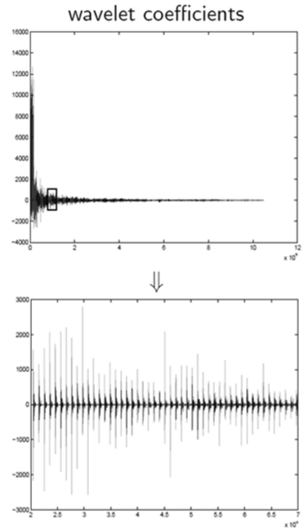
- ❑ Candes, Romberg and Tao, 2006
- ❑ Candels and Tao, 2006
- ❑ Donoho, 2006

** Emmanuel Candes, Compressive Sensing – A 25 Minute Tour*

信息科学与技术学院
School of Information Science and Technology

Compressive Sensing

1 megapixel image



信息科学与技术学院
School of Information Science and Technology

Compressive Sensing

1. Compute 1,000,000 wavelet coefficients of mega-pixel image
2. Set to zero all but the 25,000 largest coefficients
3. Invert the wavelet transform


original image


after zeroing out smallest coefficients

信息科学与技术学院
School of Information Science and Technology

Compressive Sensing

- Take 96K incoherent measurements of "compressed" image
- Compressed image is perfectly sparse (25K nonzero wavelet coeffs)
- Solve ℓ_1


original (25k wavelets)


perfect recovery

4/27/2015 -5/21/2015

Introduction to Communications

School of Information Science & Technology

ShanghaiTech University

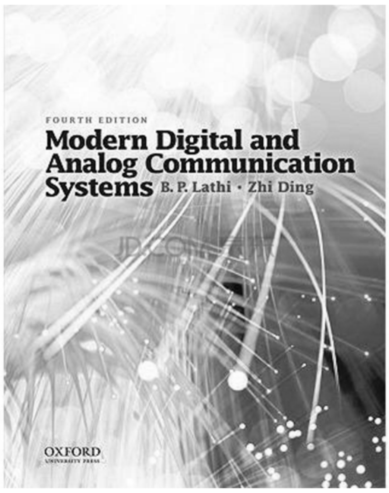
by Yanlin Geng & Xiaojun Yuan

信息科学与技术学院
School of Information Science and Technology

信息科学与技术学院
School of Information Science and Technology

References

Lathi and Ding, 4th Edition, 2009.



Word Counting

信息科学与技术学院
School of Information Science and Technology

A sequence: "ACBACABA"

Q: How many A, B, C?
A: A = 4, B = 2, C = 2

Q: How frequently?
A: pA = 0.5, pB = 0.25, pC = 0.25

Word Counting

信息科学与技术学院
School of Information Science and Technology

Use {0, 1}'s to represent A, B, C:
? minimum length
can recover "ACBACABA"

Observation:
more frequently → shorter {0, 1} sequence

Say: A → 0, B → 1, C → 01
Then: "ACBACABA" → "0011001010"

Q: can we recover "ACBACABA" uniquely?

Word Counting

信息科学与技术学院
School of Information Science and Technology

A: no, since 001 → AC or AAB

Q: how to solve?

Huffman coding

信息科学与技术学院
School of Information Science and Technology

A: Huffman coding

- A → 1
- B → 00
- C → 01

Q: any pattern?

A: prefix-free

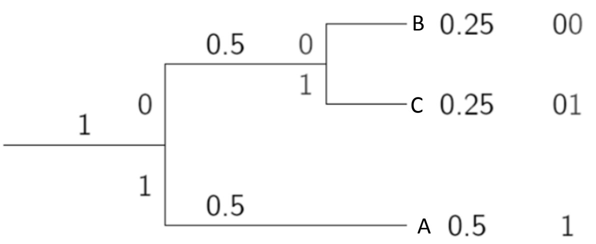
- no codeword is prefix of others
- prefix code

信息科学与技术学院
School of Information Science and Technology

Huffman coding

Q: how Huffman coding works?
A: Keep merging two smallest frequencies:

A sequence: "ACBACABA" p_i codeword

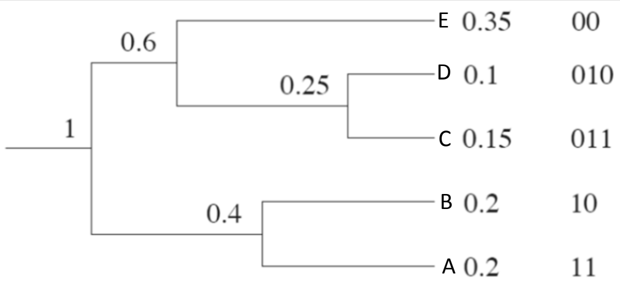


信息科学与技术学院
School of Information Science and Technology

Huffman coding

A slightly more complicated example.

“CEADBECDEEBCABEAEBAE" p_i codeword

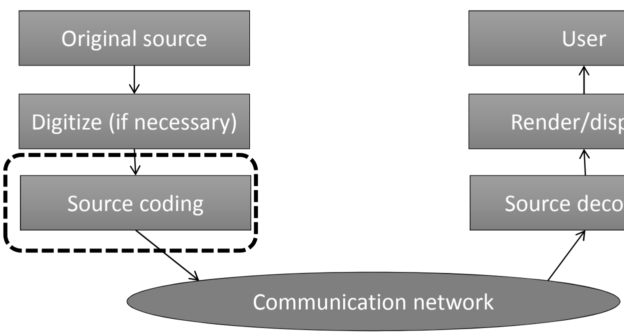


Huffman coding

Q: codewords using {0, 1, 2}? find out

信息科学与技术学院
School of Information Science and Technology

The End-to-End Commun. System



- The following lecture is about the oval.
- The simplest network is a single physical communication link.

Modern Commun. is Digitalized

Voice communication

Instant messaging

Remote file transfer

010101000111……

Everything is just bits

信息科学与技术学院
School of Information Science and Technology

Physical Links Inherently Analog

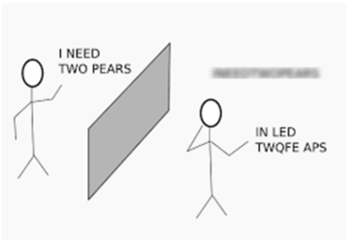
Analog = continuous-valued and continuous-time

– Voltage waveform on a cable
– Light on a fiber, or in free space
– Radio waves through the air

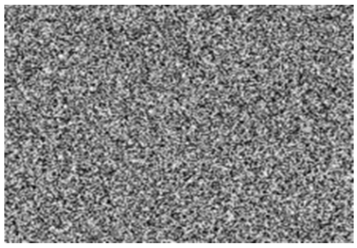
Solution → Modulation

信息科学与技术学院
School of Information Science and Technology

...and Physical Links Inherently Noisy



Obstacles

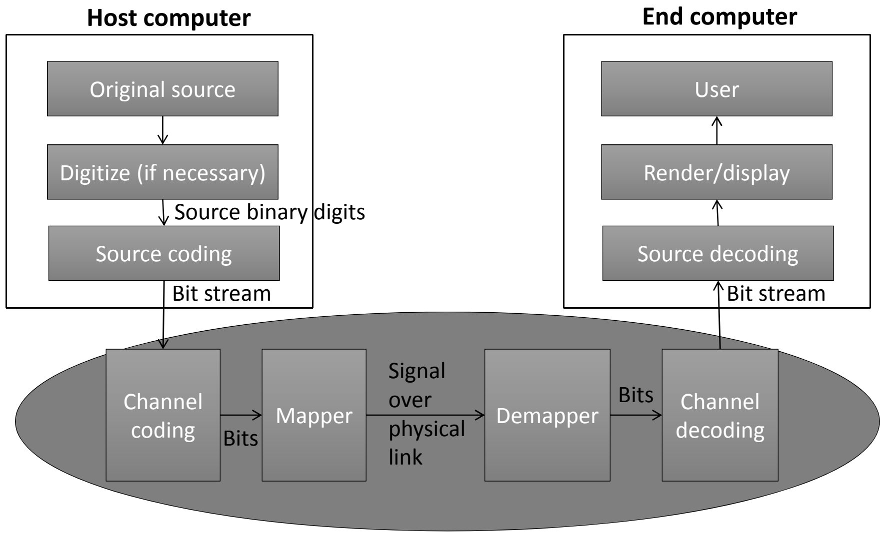


White noise

- Noise usually comes from the ambient environment and communication media.
- Communication errors occur due to channel noise.
- Solution → Channel coding

信息科学与技术学院
School of Information Science and Technology

Single-Link Communication Model



信息科学与技术学院
School of Information Science and Technology

Modulation: Map Bits to Signals

Key Idea: Map or modulate the desired bit sequence onto a (continuous-time) analog signal (waveform)

For ease of extracting the intended bits from the noisy received signals, we map bits to signals using a fixed set of discrete values.

For example, a two-level signaling scheme uses two “voltages”:

- V_0 is the binary value for “0”
- V_1 is the binary value for “1”

If $V_0 = -V_1$, we refer to this as bipolar signaling.

信息科学与技术学院
School of Information Science and Technology

Digital Signaling: Receiving

At the receiver, process and sample to get a “voltage”

- Voltages near V_0 would be interpreted as “0”
- Voltages near V_1 would be interpreted as “1”

If V_0 and V_1 are spaced far enough apart, we can tolerate some degree of noise – but there will be occasional errors!

But you receive $r = V_0 + \text{noise}$

or $r = V_1 + \text{noise}$

Digital Signaling: Receiving

We can specify the behavior of the receiver with the following decision rule (that shows how incoming voltages are mapped to “0” and “1”).

Decision Rule:

If $r<d$, then “0” is transmitted;
If $r>d$ then “1” is transmitted.

In the above, d is called the threshold voltage.

The threshold voltage is usually chosen as the middle of the two voltage levels, i.e., $d = (V_0+V_1)/2$. Why?

How to Reduce Decoding Error?

One simple way to reduce decoding error is by repetition.

Code: Bit b coded as bb…b (n times)

Exponential fall-off (log scale)

But huge overhead (low code rate)

We can do much better!

信息科学与技术学院
School of Information Science and Technology

Hamming Distance

The Hamming distance (HD) between two strings of equal length is the number of positions at which the corresponding symbols are different.

Examples:

- HD between "karolin" and "kerstin" is 3.
- HD between 1011101 and 1001001 is 2.

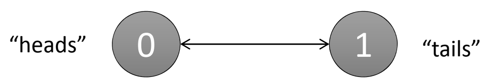
The HD between a valid binary codeword and the same codeword with e errors is e.

信息科学与技术学院
School of Information Science and Technology

Hamming Distance

The Hamming Distance (HD) between a valid binary codeword and the same codeword with e errors is e.

The problem with no coding is that the two valid codewords ("0" and "1") also have HD = 1. So a single-bit error changes a valid codeword into another valid codeword.



Q: What is the Hamming distance of a repetition code?

信息科学与技术学院
School of Information Science and Technology

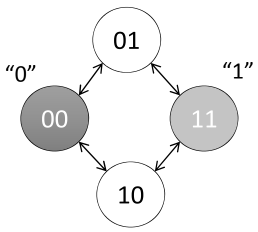
Hamming Distance and Coding

Encode so that the codewords are “far enough” from each other.

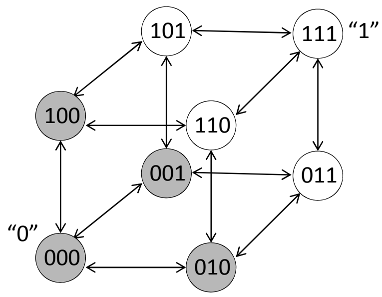
Likely error patterns shouldn’t transform one codeword to another.

Code: nodes chosen in hypercube + mapping of message bits to nodes

We choose 2k out of 2n nodes, meaning that we can map all k-bit message strings in a space of n-bit codewords. The code rate is k/n.



Can detect one-bit error



Can correct one-bit error

信息科学与技术学院
School of Information Science and Technology

Minimum HD vs. Detection & Correction Capabilities

If d is the minimum Hamming distance between codewords, we can detect all patterns of <= (d-1) bit errors

If d is the minimum Hamming distance between codewords, we can correct all patterns of (d-1)/2 or fewer bit errors

But how to construct codes with the above properties?

信息科学与技术学院
School of Information Science and Technology

A Simple Code: Parity Check

Add a parity check to message of length k to make the total number of "1" bits even.

If the number of "1"s in the received word is odd, then there has been an error.

- 010111100010 → original word with parity bit
- 010011100010 → single-bit error (detected)
- 010001100010 → two-bit error (not detected)

Minimum hamming distance of parity check code is 2.

- Can detect all single-bit errors
- Can detect all odd numbers of errors
- Cannot detect even numbers of errors
- Cannot correct any errors

信息科学与技术学院
School of Information Science and Technology

Modulo-2 Algebra

Computations with binary numbers in code construction will involve Boolean algebra, or algebra in "GF(2)" (Galois field of order 2) or modulo-2 algebra:

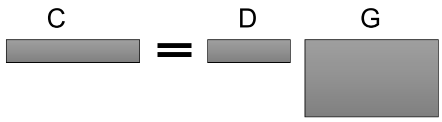
0+0 = 0, 1+0 = 0+1 = 1, 1+1 = 0

0*0 = 0*1 = 1*0 = 0, 1*1 = 1

信息科学与技术学院
School of Information Science and Technology

Linear Block Codes

Block Code: k message bits encoded to n code bits, i.e., each of 2k messages encoded into a unique n-bit combination via a linear transformation with GF(2) operations:



C is an n-element row vector containing the codeword

D is a k-element row vector containing the message

G is a k-by-n generation matrix

Each codeword bit is a specified linear combination of message bits.

信息科学与技术学院
School of Information Science and Technology

Minimum HD of Linear Code

- **Key property**: Sum of any two codewords is still a codeword → necessary and sufficient for code to be **linear**
  - So the all-zero codeword must be in any linear code --- why?
- (n, k) code has rate k/n
- Sometimes written as *(n, k, d)*, where d is the minimum Hamming distance of the code.
- The weight of a codeword is the number of “1”s in it.

- **The minimum HD of a linear code is the minimum weight found in its nonzero codewords. Why?**

Examples: What are n, k, d?

信息科学与技术学院
School of Information Science and Technology

{000, 111}	(3, 1, 3), rate = 1/3
{0000, 1100, 0011, 1111}	(4, 2, 2), rate = 1/2
{1111, 0000, 1100}	non-linear code
{0000, 1000, 0011, 1111}	non-linear code

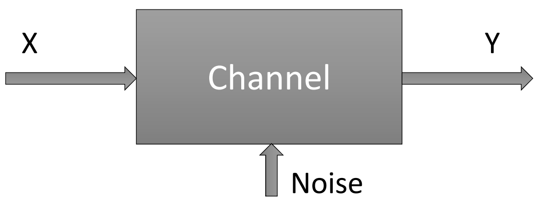
0000000	1100001	1100110	0000111	**(7, 4, 3) code, rate = 4/7**
0101010	1001011	1001100	0101101	
1010010	0110011	0110100	1010101	
1111000	0011001	0011110	1111111	

The HD of a linear code is the minimum number of 1's in all codewords.

How Fast We can Deliver Bits

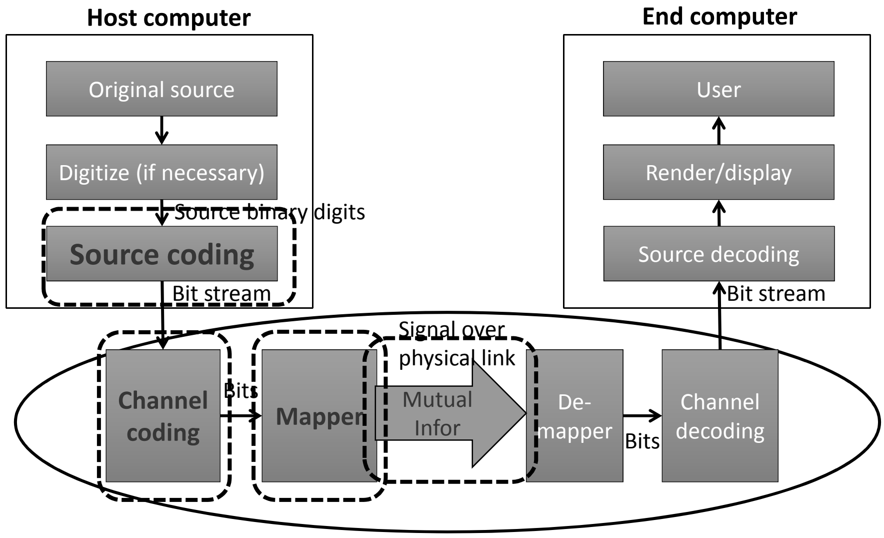
信息科学与技术学院
School of Information Science and Technology

- **H(X):**
  - **Entropy** = the uncertainty in X
- **H(X|Y):**
  - **Uncertainty** about X reduced by knowing Y
- **I(X;Y) = H(X) – H(X|Y):**
  - **Mutual information** = **RATE** supported by the channel



信息科学与技术学院
School of Information Science and Technology

Single-Link Communication Model



信息科学与技术学院
School of Information Science and Technology

Homework

1. For the following text file, find out the number of occurrences of A, B, ..., Z and the space ('a' taken as 'A', 'b' as 'B', etc.)

a). carry out the Huffman coding and calculate the number of bits needed to store this text file

b). how many bits are needed if we simply apply the ASCII coding scheme?

```
Samsung still appears to be exclusively using its eight core
Exynos five four two zero as the S six app processor Though
the WSJ does not state whether Qualcomm will do so there is a
chance the company will also supply complementary RF transceiver
power management and envelope tracking ICs in units containing
its modems as is the case with the iPhone six which relies on a
Qualcomm modem and several complementary chips to go with Apple
A eight app processor
```

信息科学与技术学院
School of Information Science and Technology

Homework

2. Calculate the HD between the following two bit sequences:

a) `1001100111000`
b) `0101110001001`

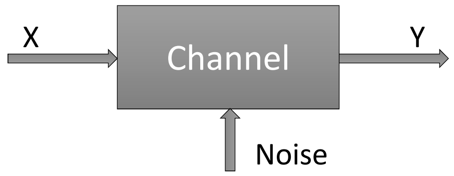
3. Generate a code, i.e., a set of codewords, with the following property:

- **4-bit codewords with three information bits and with single error detection capability i.e. min HD = 2**
- **Can we find 4-bit codewords for 3 information bits with one error correction capability? Justify your answer.**

信息科学与技术学院
School of Information Science and Technology

Advanced Reading

Channel Capacity



To characterize the channel, define

C = max {I(X; Y)} = max {H(X) – H(X|Y)}

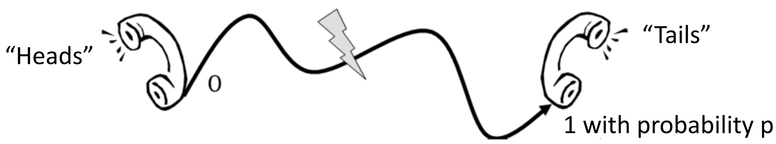
where the maximization is over all possible distributions of X.

This is the most we can expect to reduce our uncertainty about X through the knowledge of Y, and so must be the most information we can expect to send through the channel.

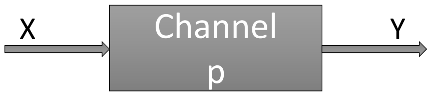
An Example: Binary Symmetric Channel

Suppose that during transmission a “0” is turned into a “1” or a “1” is turned into a “0” with probability *p*, independently of transmissions at other times.

This is a binary symmetric channel (BSC) – a useful and widely used abstraction.

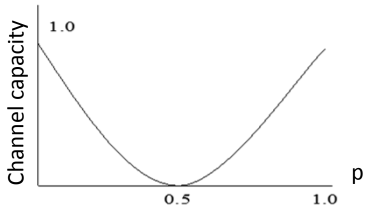


Capacity of the Binary Symmetric Channel



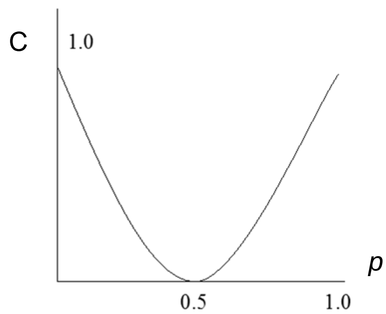
C = max {H(Y) – H(Y|X)}, where the maximization is over all possible distributions of X.

The second term doesn't depend on this distribution, and the first term is maximized when 0 and 1 are equally probable at the input.



信息科学与技术学院
School of Information Science and Technology

Capacity for Binary Symmetric Channel



For low noise channel, significant reduction in uncertainty about the input after observing the output. Explain why C = 1 for p = 0.

For high noise channel, little reduction. Explain why C = 0 for p = 0.5.

What happens to p = 1?

信息科学与技术学院
School of Information Science and Technology

Channel capacity tells us how fast and how accurately we can communicate …

信息科学与技术学院
School of Information Science and Technology

Magic of Asymptotically Error-Free if R < C

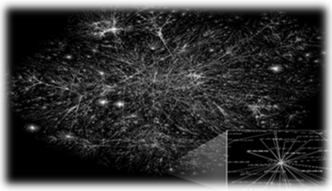
Shannon showed that one can theoretically transmit information (i.e., message bits) at any rate $R < C$ per use of the channel, with arbitrarily low error.

He also showed the converse, that transmission at any average rate $R \geq C$ incurs an error probability that is low-bounded by some positive number.

The secret: Encode blocks of k message bits into n-bit codewords, so $R = k/n$, with k and n very large.

Encoding blocks of k message bits into n-bit codewords to protect against channel errors is an example of channel coding.

4/27/2015 -5/21/2015



Introduction to Networks

School of Information Science & Technology
ShanghaiTech University

by Ziyu Shao

信息科学与技术学院
School of Information Science and Technology

信息科学与技术学院
School of Information Science and Technology

Reference

Acknowledgement:
Main contents of Parts 2,3,4 adopt materials from the official slides to accompany the following book, which is also the reference book

Computer Networking: A Top Down Approach

6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

信息科学与技术学院
School of Information Science and Technology

Outline

1 Complex Networks
2 Foundation of Internet
3 Performance Metrics of Networks
4 Protocols & Layers
5 History

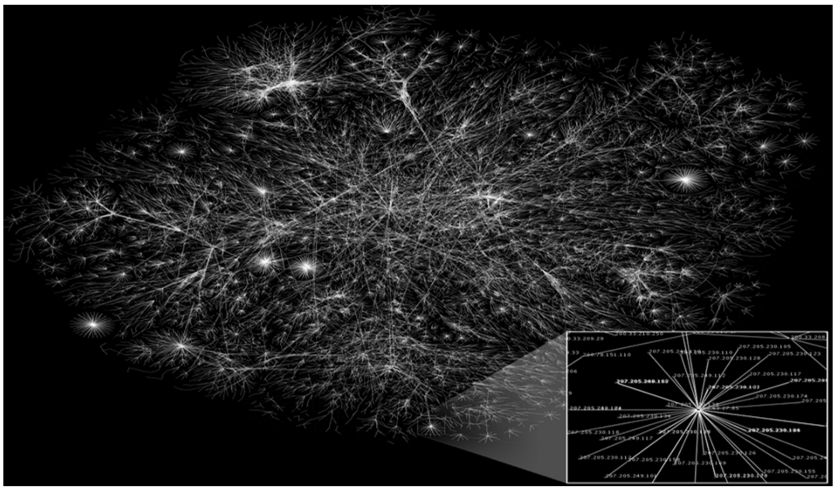
信息科学与技术学院
School of Information Science and Technology

Outline

1 Complex Networks
2 Foundation of Internet
3 Performance Metrics of Networks
4 Protocols & Layers
5 History

信息科学与技术学院
School of Information Science and Technology

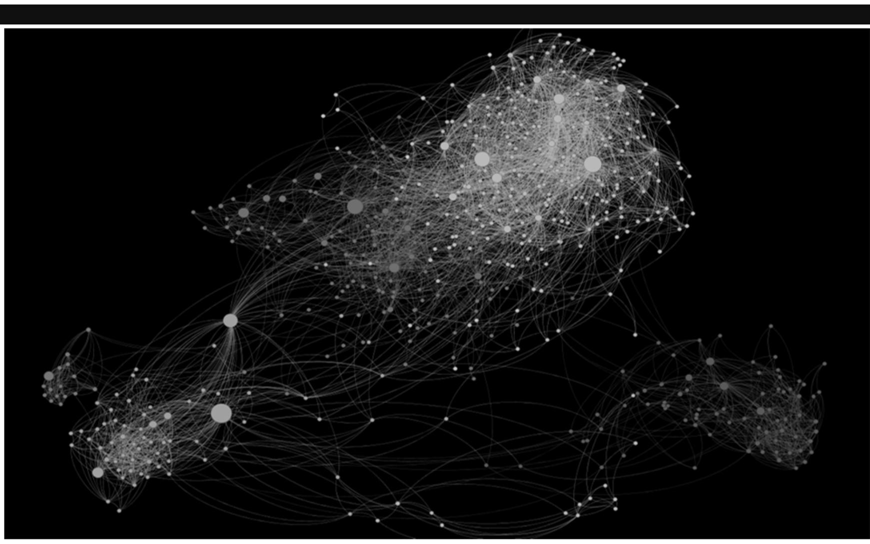
World Wide Web(WWW)



Source: http://en.wikipedia.org/wiki/File:Internet_map_1024.jpg

信息科学与技术学院
School of Information Science and Technology

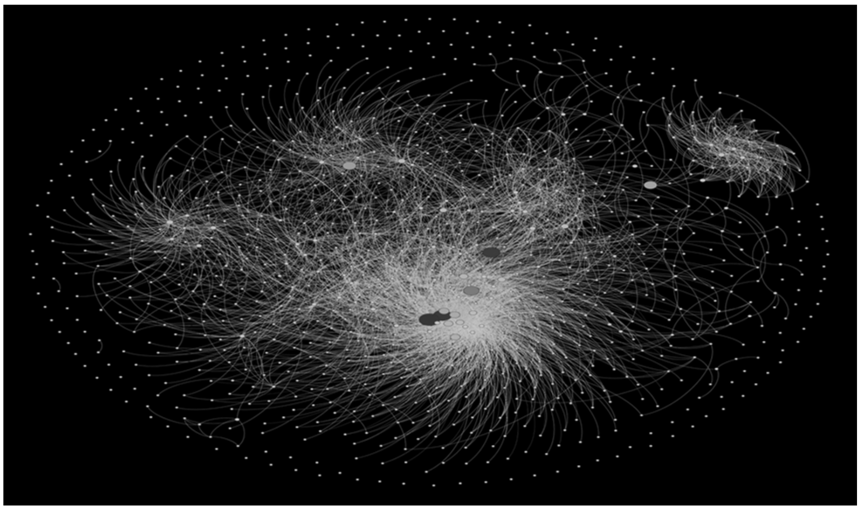
Facebook Friendship Network



Source: http://kimoquaintance.com/2011/08/22/what-can-we-learn-about-somalis-from-their-facebook-networks/

信息科学与技术学院
School of Information Science and Technology

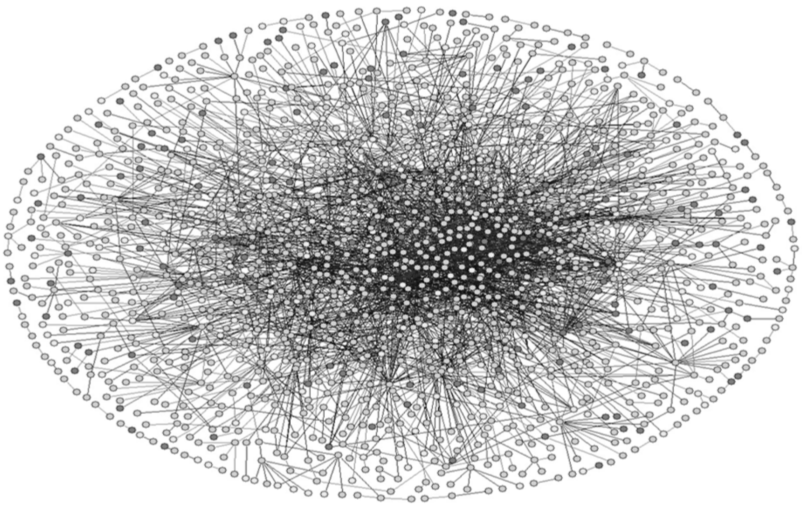
Facebook Friendship Network



Source: http://kimoquaintance.com/2011/08/22/what-can-we-learn-about-somalis-from-their-facebook-networks/

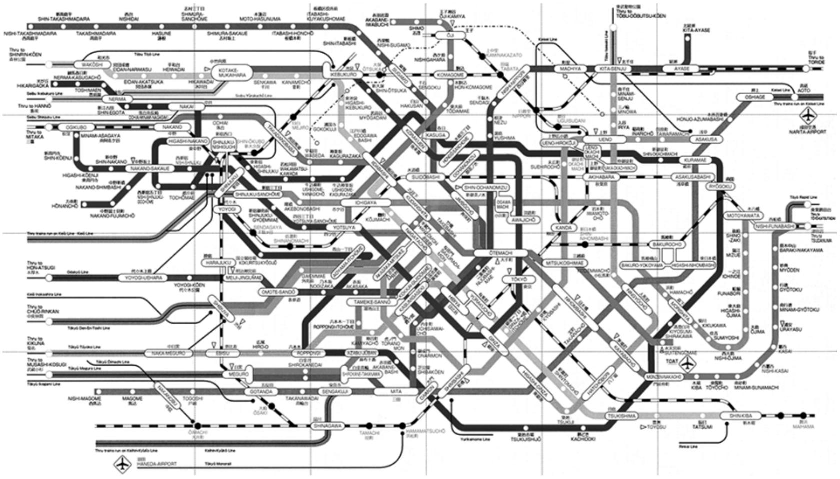
信息科学与技术学院
School of Information Science and Technology

Protein Interaction Networks



Source: https://www.mdc-berlin.de/16074534/en/news/archive/2008/20080910-erwin_schr_dinger_prize_2008_goes_to_resea

Tokyo Transportation Network



Source: https://maximizetravel.wordpress.com/2012/01/24/tokyo-transportation

信息科学与技术学院
School of Information Science and Technology

Outline

1 Complex Networks
2 Foundation of Internet
3 Performance Metrics of Networks
4 Protocols & Layers
5 History

信息科学与技术学院
School of Information Science and Technology

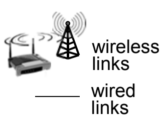
Internet Topology

Source: http://www.jacobsschool.ucsd.edu/news/news_releases/release.sfe?id=685

信息科学与技术学院
School of Information Science and Technology

Elements of Internet

Millions of connected computing devices (*hosts*)

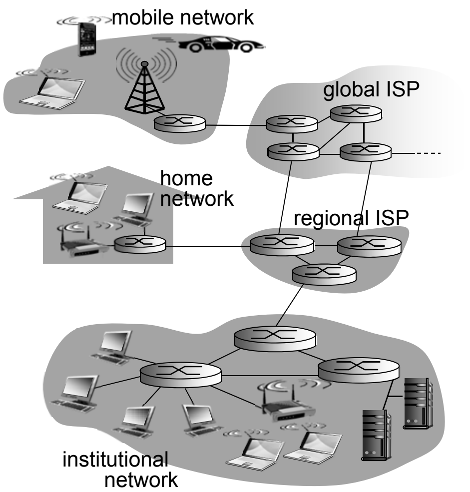


Communication Links (*optical fiber, copper, radio, satellite*)

Packet Switches (*routers & switches*)

信息科学与技术学院
School of Information Science and Technology

Structure of Internet



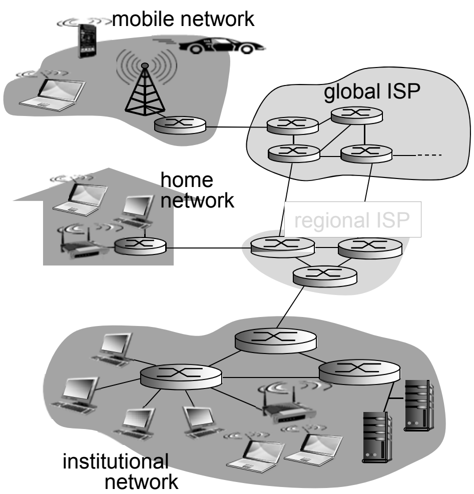
Network Edge:
end systems with hosts & access networks

Access Network:
connect end systems to edge routers

Network Core:
interconnected routers
network of networks

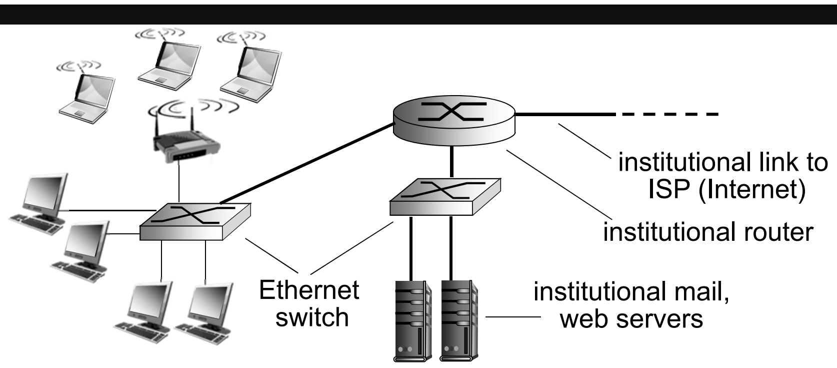
信息科学与技术学院
School of Information Science and Technology

Access Network



The Last Mile Problem:
bandwidth (bits per second) of access network

Enterprise Access Network (Ethernet)



Typically used in companies, universities, etc

10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates

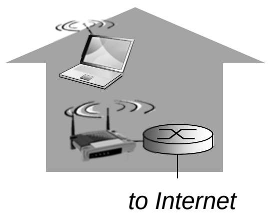
End systems typically connect into Ethernet switch

Wireless Access Network

- Shared *wireless* access network connects end system to router
  - via base station aka "access point"

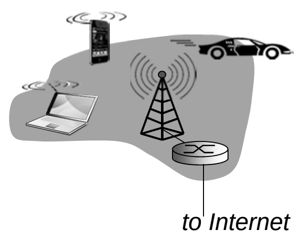
wireless LANs:

- within building (100 ft)
- 802.11b/g (WiFi): 11, 54 Mbps transmission rate



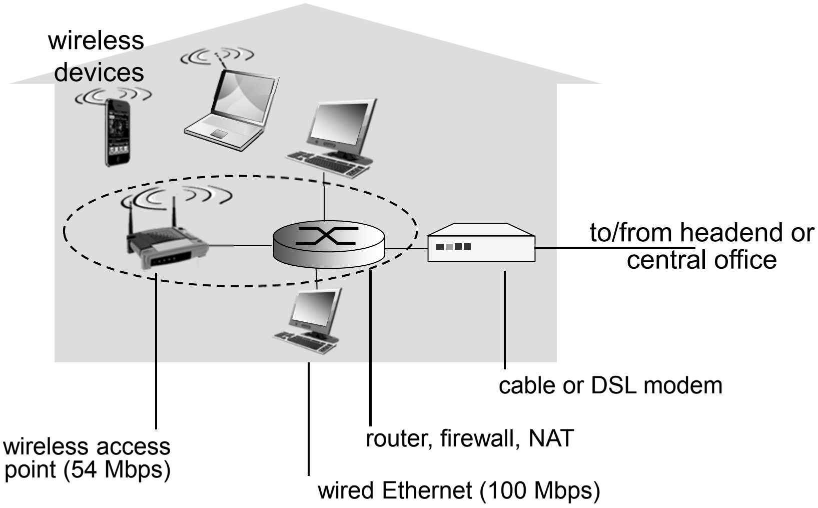
wide-area wireless access

- provided by telco (cellular) operator, 10's km
- between 1 and 10 Mbps
- 3G, 4G: LTE



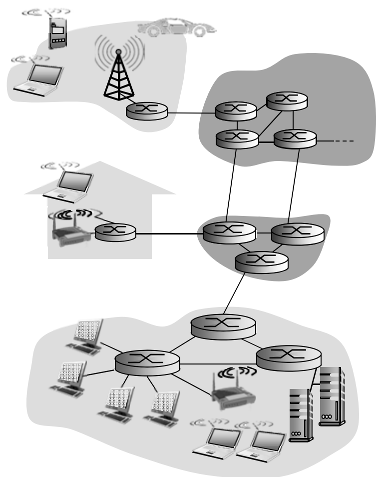
信息科学与技术学院
School of Information Science and Technology

Home Access Network



信息科学与技术学院
School of Information Science and Technology

Network Core



Mesh of interconnected routers

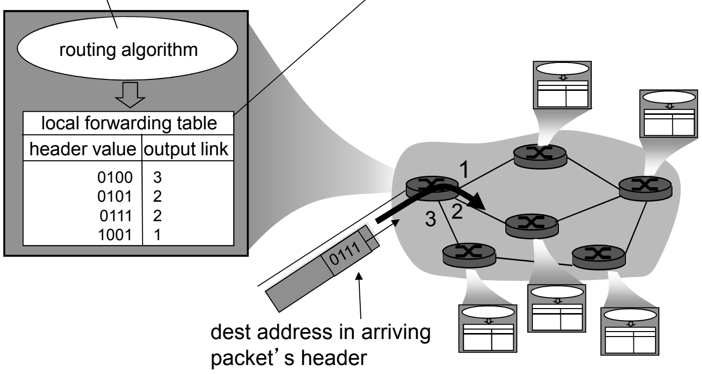
Packet Switching:

- *Hosts break messages into packets*
- *Forward packets from one router to the next, across links on path from source to destination*
- *Each packet transmitted at full link capacity*

Two Key Functions of Router

routing: determines source-destination route taken by packets
- *routing algorithms*

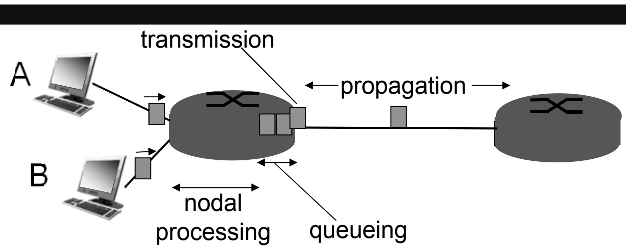
forwarding: move packets from router's input to appropriate router output



Outline

1 Complex Networks
2 Foundation of Internet
3 Performance Metrics of Networks
4 Protocols & Layers
5 History

Metric 1: Packet Delay

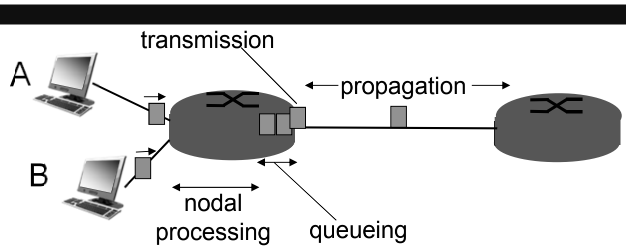


$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$

d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router

Metric 1: Packet Delay



$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$

d_{trans}: transmission delay:
- *L*: packet length (bits)
- *R*: link *bandwidth (bps)*
- $d_{trans} = L/R$

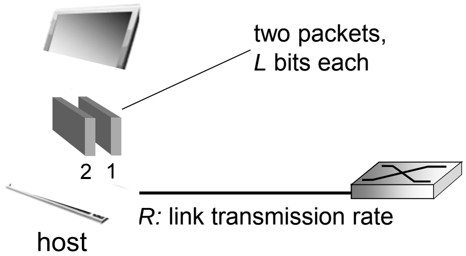
d_{prop}: propagation delay:
- *d*: length of physical link
- *s*: propagation speed in medium (~3×10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop} *very* different

信息科学与技术学院
School of Information Science and Technology

Transmission Delay: Scenario 1

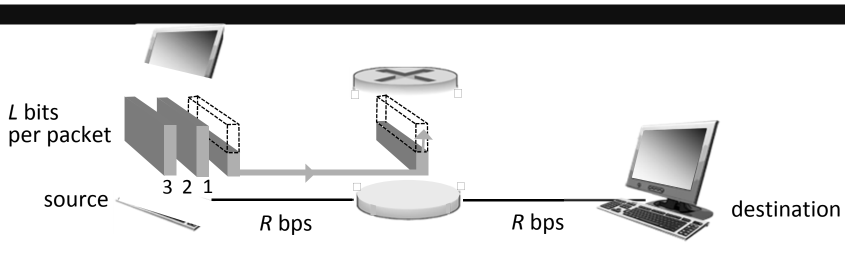
host sending function:
- takes application message
- breaks into smaller chunks, known as *packets*, of length *L* bits
- transmits packet into access network at *transmission rate R*
  - link transmission rate, aka link *capacity, aka link bandwidth*



packet transmission delay = time needed to transmit *L*-bit packet into link = $\frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$

信息科学与技术学院
School of Information Science and Technology

Transmission Delay : Scenario 2



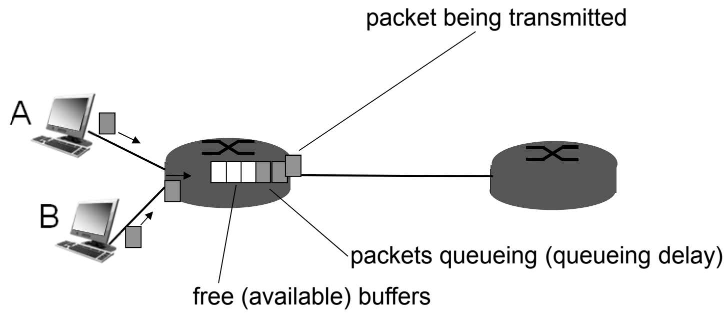
- takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link

one-hop numerical example:
- *L* = 7.5 Mbits
- *R* = 1.5 Mbps
- one-hop transmission delay = 5 sec

Queueing Delay

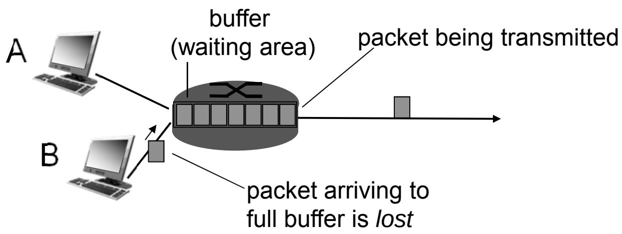
packets *queue* in router buffers

- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



Metric 2: Packet Loss

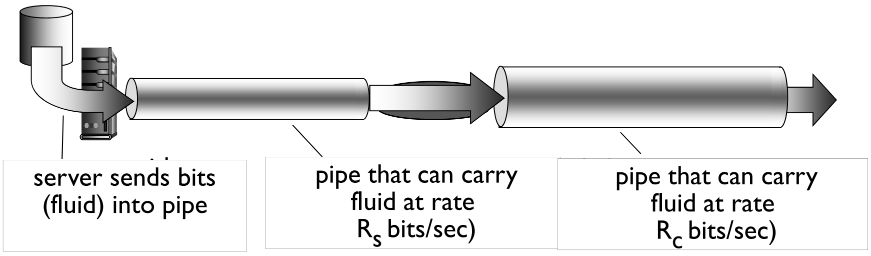
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source of end system, or not at all



信息科学与技术学院
School of Information Science and Technology

Metric 3: Throughput

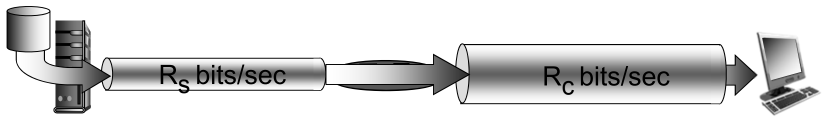
- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time



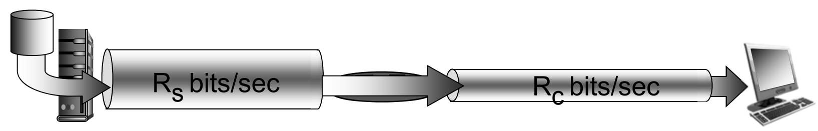
信息科学与技术学院
School of Information Science and Technology

Throughput (more)

- $R_s < R_c$ What is average end-end throughput?



- $R_s > R_c$ What is average end-end throughput?



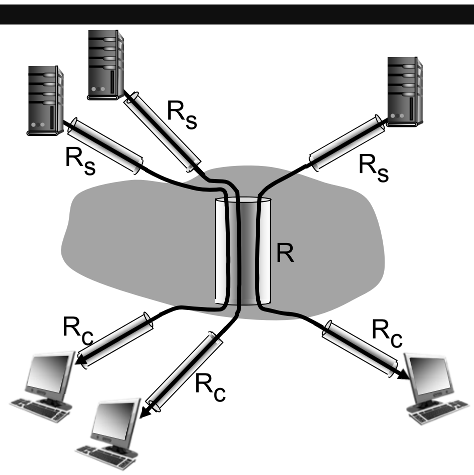
bottleneck link

link on end-end path that constrains end-end throughput

信息科学与技术学院
School of Information Science and Technology

Throughput: Internet Scenario

- per-connection end-end throughput: min(R_c,R_s,R/10)
- in practice: R_c or R_s is often bottleneck



10 connections (fairly) share backbone bottleneck link R bits/sec

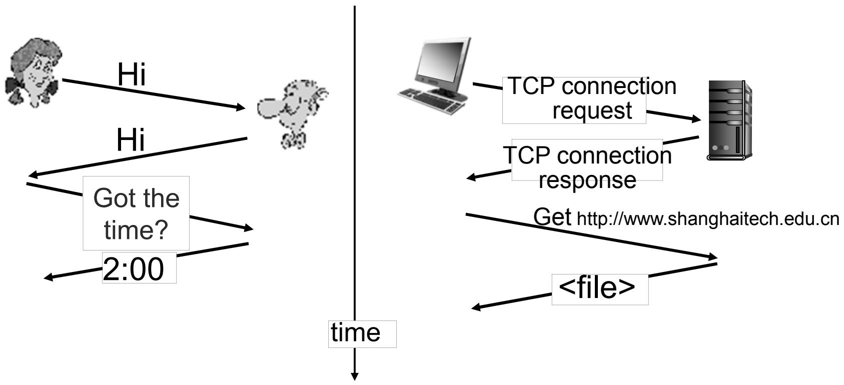
信息科学与技术学院
School of Information Science and Technology

Outline

信息科学与技术学院
School of Information Science and Technology

Protocols: Laws of Networks

a human protocol vs. a computer network protocol:



信息科学与技术学院
School of Information Science and Technology

Protocols: Laws of Networks

Protocols define format, order of messages sent and received among network entities, and actions taken on message transmission & receipt

All communication activity in Internet governed by protocols

Examples: TCP, UDP, IP, BGP, HTTP, 802.11

The Big Question

Networks are complex, with many "pieces":

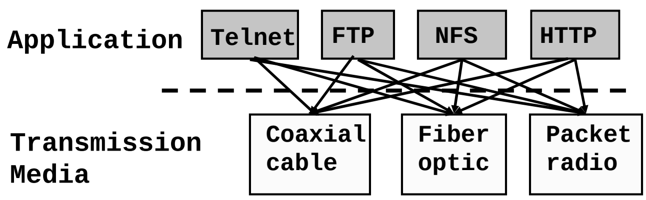
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

is there any hope of *organizing* structure of network?

信息科学与技术学院
School of Information Science and Technology

The Problem



- Do we re-implement every application for every technology?
- Obviously not, but how does the Internet architecture avoid this?

信息科学与技术学院
School of Information Science and Technology

Architecture

- Architecture is not the implementation itself
- Architecture is how to “organize” implementations
  - what interfaces are supported
  - where functionality is implemented
- Architecture is the modular design of the network

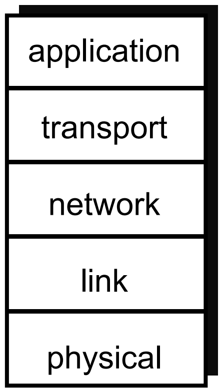
信息科学与技术学院
School of Information Science and Technology

Layering

- Layering is a particular form of modularization
- The system is broken into a vertical hierarchy of logically distinct entities (layers)
- The service provided by one layer is based solely on the service provided by layer below
- Rigid structure: easy reuse, performance may suffers

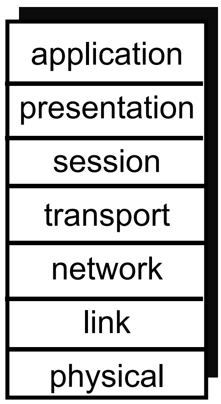
Layering Model for Internet

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi)
- *physical:* bits "on the wire"



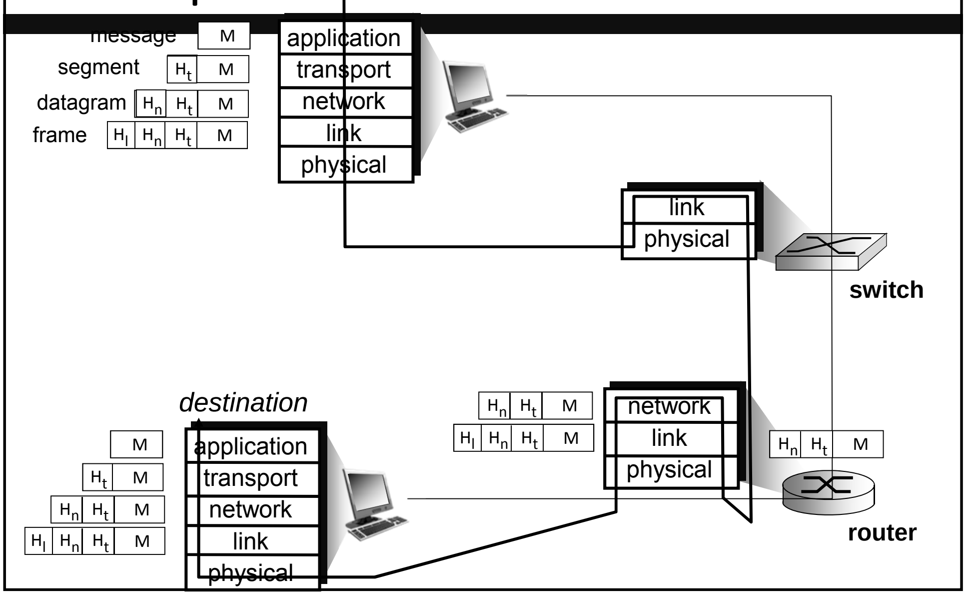
OSI Reference Model for Layers

- *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
  - these services, *if needed,* must be implemented in application
  - needed?



信息科学与技术学院
School of Information Science and Technology

Encapsulation



信息科学与技术学院
School of Information Science and Technology

Layering Solves Problem

- Application layer doesn't know about anything below the presentation (or transport) layer, etc.
- Information about network is hidden from higher layers
- This ensures that we only need to implement an application once!

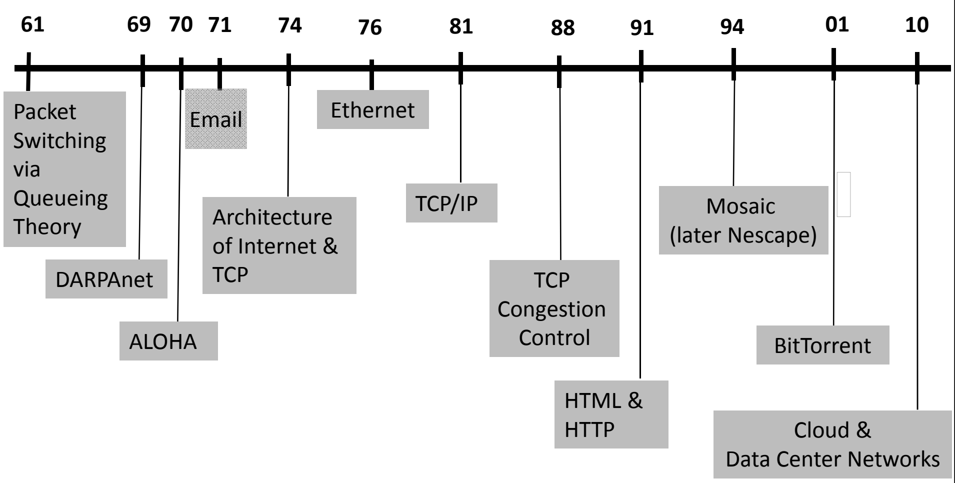
信息科学与技术学院
School of Information Science and Technology

Outline

1 Complex Networks
2 Foundation of Internet
3 Performance Metrics of Networks
4 Protocols & Layers
5 History

信息科学与技术学院
School of Information Science and Technology

History of Moments



信息科学与技术学院
School of Information Science and Technology

Father of the Internet

Vincent Cerf

Robert Kahn

Interesting video: *https://www.youtube.com/watch?v=xA6Ccg4sdXc*

2014 public lecture by Vincent Cerf & Robert Kahn for the 40th anniversary of Internet.

信息科学与技术学院
School of Information Science and Technology

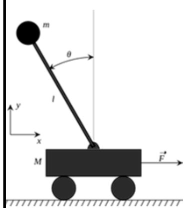
Homework: Simulation

1. Consider a sequence of packets passing through one link.

- Packets arrive one by one, and the time intervals between arriving packets satisfying a probability distribution with a mean equaling to 6 units.
- The bandwidth of the link is random, satisfying a probability distribution with a mean equaling to 0.2 packet per units.
- The size of the link buffer is 3(full buffer with three packets).
- Zero nodal processing delay & propagation delay.

You need to use python to simulate the behavior of such system with various probability distributions. Please record the average queueing delay, average delay, average throughput and loss rate. You are also required to make observations and provide intuitive explanations.

4/27/2015 -5/21/2015



Feedback Control

School of Information Science & Technology
ShanghaiTech University

by Boris Houska

信息科学与技术学院
School of Information Science and Technology

Overview

Why do we need control systems ?

What is a feedback controller ?

How do modern feedback controllers work ?

Can you stand on one leg?

Can you stand on one leg?

Try it …

Now close your eyes…

What did you experience?

Why is it difficult to keep your balance with closed eyes?

Basic Question of Control

Sensors ("Eyes")
- Cameras
- Radar
- GPS
- Inertial sensors (IMU)
- Pressures
- Temperatures
- ...

Basic Question of Control

Sensors ("Eyes")
- Cameras
- Radar
- GPS
- Inertial sensors (IMU)
- Pressures
- Temperatures
- ...

Actuators ("Muscles")
- Motors
- Flaps
- Valves
- Propellers
- Heaters
- Pumps
- ...

Basic Question of Control

Sensors ("Eyes")
- Cameras
- Radar
- GPS
- Inertial sensors (IMU)
- Pressures
- Temperatures
- ...

Actuators ("Muscles")
- Motors
- Flaps
- Valves
- Propellers
- Heaters
- Pumps
- ...

How to connect?

Feedback Control

Step 1: Wait for the measurement from the sensor

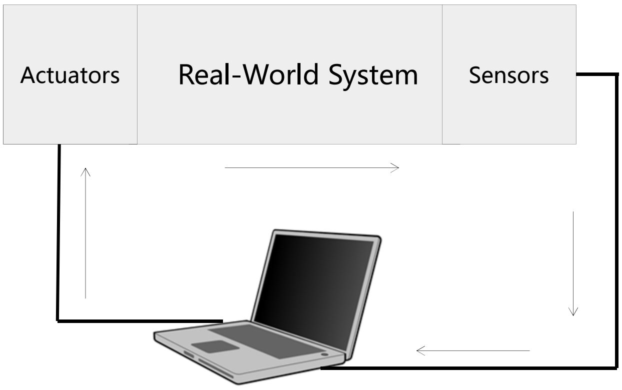
Feedback Control

Step 2: Filter the incoming data and compute a control reaction

Feedback Control

Step 3: Send out the control signal to the actuators

Feedback Control



Run Step 1 – 3 in a loop!

Zillions of Applications

Airplanes start and land using auto-pilots

Most chemical production process are optimized and automated

Automatic heating control in building saves huge amounts
of energy

… there are many many more !!!

(source: wikipedia)

Quiz

What are the actuators?

Quiz

What are the sensors?

Quiz

What are the actuators?

Quiz

What are the sensors?

Quiz

What are the actuators?

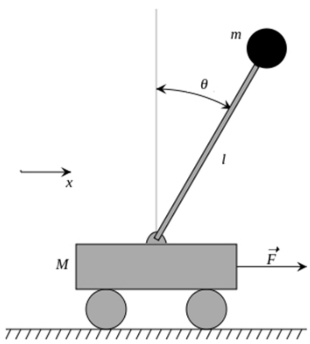
Quiz

What are the sensors?

Goal of this lecture

Learn how to design a controller with a few lines of Python code

Motivating Example: Inverted Pendulum



Can we bring the pendulum to its inverted position?

Sensor: Camera measuring positions and velocities

Actuator: motor adjusting the force F

Potential and Kinetic Energy

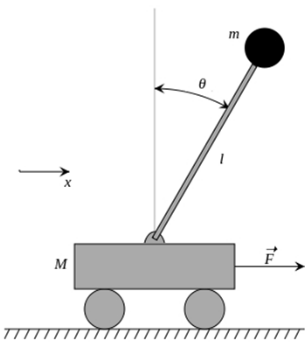
Potential energy:

$$V = mgh$$

Kinetic energy:

$$T = \frac{1}{2}mv^2$$

Energy of the Inverted Pendulum



Potential energy:

$$V = mgl\cos(\theta)$$

Kinetic energy:

$$T = \frac{1}{2}M\dot{x}^2 + \frac{m}{2}\left(\dot{x}^2 + 2l\dot{x}\dot{\theta}\cos(\theta) + l^2\dot{\theta}^2\right)$$

Notation: dot above variable means derivative w.r.t. time

Equations of Motion

Lagrange function:

$$L = T - V$$

Lagrange formalism:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{x}} - \frac{\mathrm{d}L}{\mathrm{d}x} = (M+m)\ddot{x} + ml\cos(\theta)\ddot{\theta} - ml\sin(\theta)\dot{\theta}^2 = F$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}} - \frac{\mathrm{d}L}{\mathrm{d}\theta} = ml^2\ddot{\theta} + ml\cos(\theta)\ddot{x} + mgl\sin(\theta) = 0$$

Equations of Motion

Lagrange function:

$$L = T - V$$

Lagrange formalism:

Inertia

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{x}} - \frac{\mathrm{d}L}{\mathrm{d}x} = (M+m)\ddot{x} + ml\cos(\theta)\ddot{\theta} - ml\sin(\theta)\dot{\theta}^2 = F$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}} - \frac{\mathrm{d}L}{\mathrm{d}\theta} = ml^2\ddot{\theta} + ml\cos(\theta)\ddot{x} + mgl\sin(\theta) = 0$$

Equations of Motion

Lagrange function:

$$L = T - V$$

Lagrange formalism:

Inertia — centrifugal force

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{x}} - \frac{\mathrm{d}L}{\mathrm{d}x} = (M+m)\ddot{x} + ml\cos(\theta)\ddot{\theta} - ml\sin(\theta)\dot{\theta}^2 = F$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}} - \frac{\mathrm{d}L}{\mathrm{d}\theta} = ml^2\ddot{\theta} + ml\cos(\theta)\ddot{x} + mgl\sin(\theta) = 0$$

Equations of Motion

Lagrange function:

$$L = T - V$$

Lagrange formalism:

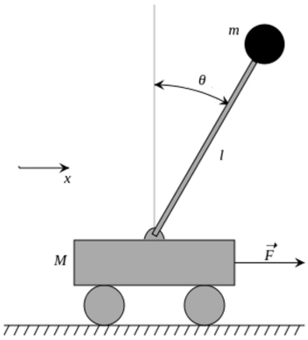
Inertia — centrifugal force

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{x}} - \frac{\mathrm{d}L}{\mathrm{d}x} = (M+m)\ddot{x} + ml\cos(\theta)\ddot{\theta} - ml\sin(\theta)\dot{\theta}^2 = F$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}} - \frac{\mathrm{d}L}{\mathrm{d}\theta} = ml^2\ddot{\theta} + ml\cos(\theta)\ddot{x} + mgl\sin(\theta) = 0$$

gravity

Simplified Equations of Motion



For small excitation angles:

$$\sin(\theta) \approx \theta \quad \cos(\theta) \approx 1 \quad \dot{\theta}^2 \approx 0$$

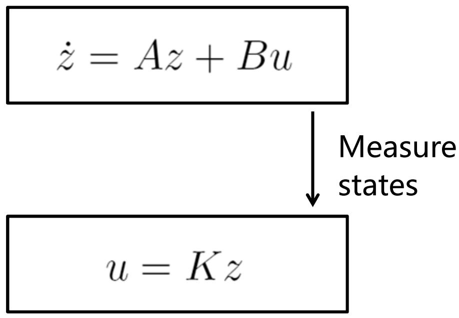
$$\begin{aligned} \ddot{x} &= \frac{F}{M} - \frac{mg}{M}\theta \\ \ddot{\theta} &= -\frac{F}{Ml} + \frac{M+m}{M}\frac{g}{l}\theta \end{aligned}$$

Linear Control System

$$\dot{z} = Az + Bu$$

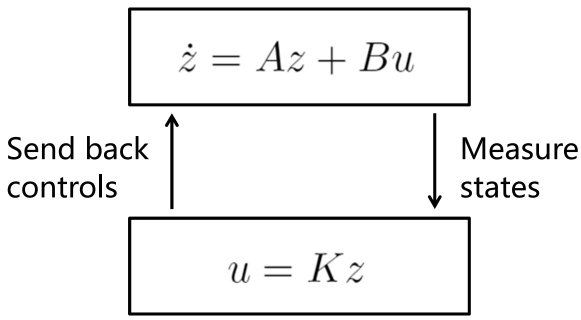
Measure states

Linear Control System



Linear feedback law

Linear Control System



Linear feedback law

Inverted Pendulum in Standard Form

States:

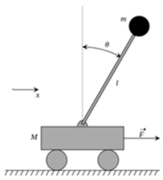
$$z = \begin{pmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{pmatrix}$$

- x: Position of trolley
- θ: Angle of the pendulum
- $\dot{x}$: Velocity of the trolley
- $\dot{\theta}$: Angular velocity of the pendulum

System matrices:

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{mg}{M} & 0 & 0 \\ 0 & \frac{M+m}{M}\frac{g}{l} & 0 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{M} \\ -\frac{1}{Ml} \end{pmatrix}$$

Linear Control System



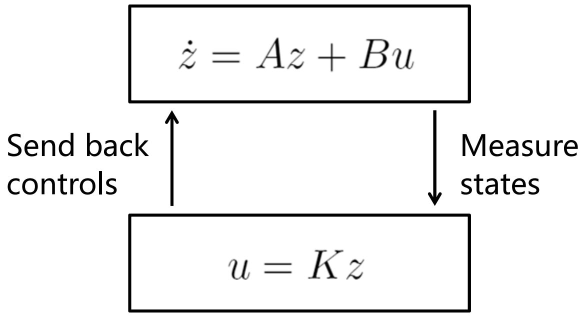
$$\dot{z} = Az + Bu$$

Send back controls ↑ ↓ Measure states

$$u = Kz$$

Linear feedback law

Linear Control System



Linear feedback law

Closed Loop Dynamics: $\dot{z} = (A + BK)\, z$

Closed Loop Dynamics in the State Space

Closed-loop trajectory satisfies

$$\dot{z} = (A + BK)\, z$$

Explicit solution

$$z(t) = e^{(A+BK)t} z(0)$$

Simulation of Closed-Loop Dynamics in Python

Math-Syntax:

$$z(t) = e^{(A+BK)t} z(0)$$

Python Syntax:

```
z = la.expm((A+B*K)*t)*z0;
```


Simulation of Closed-Loop Dynamics in Python

Math-Syntax:

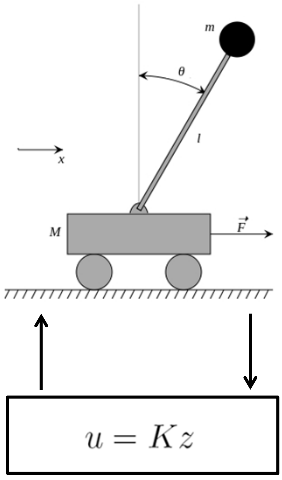
$$z(t) = e^{(A+BK)t} z(0)$$

Python Syntax:

```
z = la.expm((A+B*K)*t)*z0;
```


Simulates linear closed-loop systems with 1 line of code!

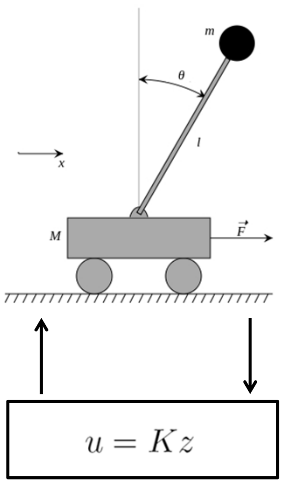
What happens for K = 0 ?



Linear feedback law

$$\ddot{x} = \frac{F}{M} - \frac{mg}{M}\theta$$
$$\ddot{\theta} = -\frac{F}{Ml} + \frac{M+m}{M}\frac{g}{l}\theta$$

What happens for K = 0 ?



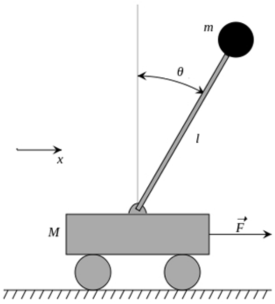
Linear feedback law

$$\ddot{x} = \frac{F}{M} - \frac{mg}{M}\theta$$
$$\ddot{\theta} = -\frac{F}{Ml} + \frac{M+m}{M}\frac{g}{l}\theta$$

For K = 0 we apply no force
u = F = 0

unstable!

How to choose K ?



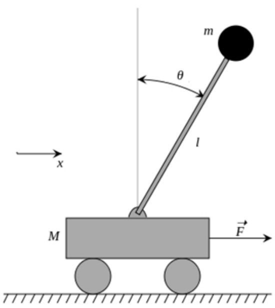
$$\ddot{x} = \frac{F}{M} - \frac{mg}{M}\theta$$
$$\ddot{\theta} = -\frac{F}{Ml} + \frac{M+m}{M}\frac{g}{l}\theta$$

Intuition:

If angle positive, choose F positive

If angle negative, choose F negative

How to choose K ?



$$\ddot{x} = \frac{F}{M} - \frac{mg}{M}\theta$$
$$\ddot{\theta} = -\frac{F}{Ml} + \frac{M+m}{M}\frac{g}{l}\theta$$

Intuition:

Let's try the choice $K = (\, 0,\ b,\ 0,\ 0\,)$ with $b > (m+M)g$

Associated control law: $F = Kz = b\theta$

Let's implement this!

```
def simulate( x0, K ):    # inputs: initial values, control gain

       import numpy        as np
       import scipy.linalg as la

       M = 0.4;               # mass of the trolley (in kg)
       m = 0.1;               # mass of the pendulum (in kg)
       l   = 0.2;               # length of the pendulum (in m)
       g  = 9.81;             # gravitational constant (in m/s^2)
```


Let's implement this!

```
def simulate( x0, K ):    # inputs: initial values, control gain

       import numpy        as np
       import scipy.linalg as la

       M = 0.4;               # mass of the trolley (in kg)
       m = 0.1;               # mass of the pendulum (in kg)
       l   = 0.2;               # length of the pendulum (in m)
       g  = 9.81;             # gravitational constant (in m/s^2)
```


Guidelines:

- Give intuitive names to variables and functions

- Don't forget to comment your code

Let's implement this!

```
def simulate( x0, K ):    # inputs: initial values, control gain

  # MODEL PARAMETERS
  # ---------------------------------------------------------
     [...]

  # SETUP OF THE SYSTEM MATRICES
  # ---------------------------------------------------------
    A = np.matrix(np.zeros((4,4)));    # initialize matrix A
    B = np.matrix(np.zeros((4,1)));    # initialize matrix B

    A[0,2] = 1.0  ;# derivative of trolley position equals velocity
    A[1,3] = 1.0  ;# derivative of angle equals angular velocity
    A[2,1] = -m*g/M ; # accelaration of the trolley
    A[3,1] = (m+M)*g/(M*l);    # angular acceleration

    B[2,0] = 1.0/M     ; # influence of the force on acceleration
    B[3,0] = -1.0/(M*l); # influence of the force on the angular
                           # acceleration
```

Let's implement this!

```
def simulate( x0, K ):   # inputs: initial values, control gain

  # MODEL PARAMETERS
  # ---------------------------------------------------------
     [...]

  # SETUP OF THE SYSTEM MATRICES
  # ---------------------------------------------------------
     [...]

  ## SIMULATION OF THE CLOSED-LOOP SYSTEM
  ## ----------------------------------------------
       N = 500 ;  # plot resolution
       T  = 10.0; # time horizon (in seconds)
       t   = np.linspace(0,T,N+1);        # setup time points
       x  = np.matrix(np.zeros((N+1,4))); # time series of states
       x[0] = x0;                         # store the initial value

       X = la.expm( (A+B*K)*(T/N) );      # state transition matrix

       for i in range(N):
           x[i+1] = x[i]*np.transpose(X);   # closed-loop simulation
```

Let's implement this!

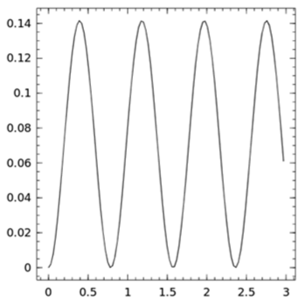
```
def simulate( x0, K ):    # inputs: initial values, control gain

  # MODEL PARAMETERS
  # ----------------------------------------------------------
     [...]

  # SETUP OF THE SYSTEM MATRICES
  # ----------------------------------------------------------
     [...]

  # SIMULATION OF THE CLOSED-LOOP SYSTEM
  # ----------------------------------------------------------
     [...]

  # RETURN THE RESULT OF THE SIMULATION
  # ----------------------------------------------------------
        return (t,x);
```

20 lines of self contained code to simulate a closed-loop system for an inverted pendulum

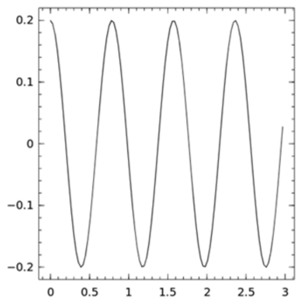
Closed-Loop Simulation Example

$$K = (\,0,\, b,\, 0,\, 0\,) \qquad b > (m+M)g$$

Trolley position [m] versus time [s]



Angle [rad] versus time [s]



Pendulum does not fall down, but oscillates

How to choose K ?

$$\begin{aligned}\ddot{x} &= \frac{F}{M} - \frac{mg}{M}\theta \\ \ddot{\theta} &= -\frac{F}{Ml} + \frac{M+m}{M}\frac{g}{l}\theta\end{aligned}$$

Intuition:

We want to "reduce" the oscillation

Strategy:

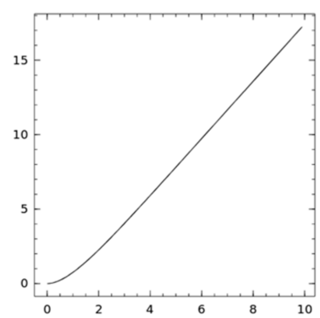
If angular velocity positive, increase F

If angular velocity negative, decrease F

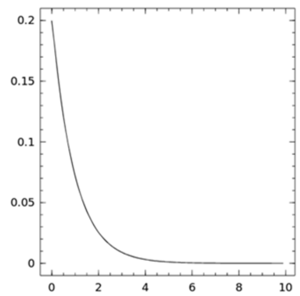
Closed-Loop Simulation Example

$$K = (\,0,\, b,\, 0,\, d\,) \qquad b > (m+M)g \qquad d > 0$$

Trolley position [m] versus time [s]



Angle [rad] versus time [s]



Pendulum stabilized at inverted position, but trolley drifts away

How to choose K ?

$$\begin{aligned}\ddot{x} &= \frac{F}{M} - \frac{mg}{M}\theta \\ \ddot{\theta} &= -\frac{F}{Ml} + \frac{M+m}{M}\frac{g}{l}\theta\end{aligned}$$

Goal of our lab exercise:

We do not only want to stabilize the pendulum, but also the trolley.

How to choose K ?

$$\begin{aligned}\ddot{x} &= \frac{F}{M} - \frac{mg}{M}\theta \\ \ddot{\theta} &= -\frac{F}{Ml} + \frac{M+m}{M}\frac{g}{l}\theta\end{aligned}$$

Goal of our lab exercise:

We do not only want to stabilize the pendulum, but also the trolley.

Strategy:

$$K = (\,a,\; b,\; c,\; d\,)$$

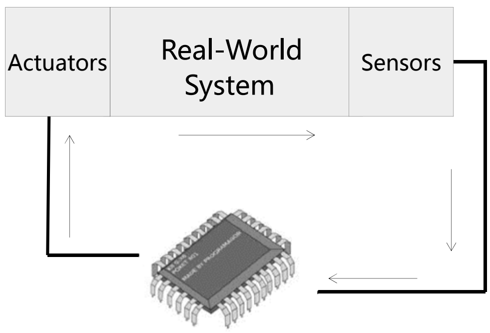
Tune all four coefficients!

Modern Feedback Control

- Nowadays, we do not tune controllers “by hand”
- Instead: use state-of-the-art optimization algorithms

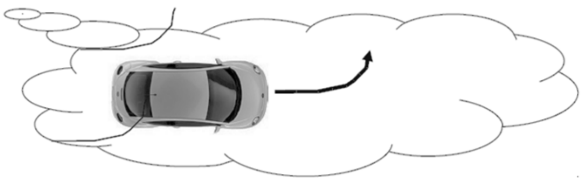
Modern Feedback Control

- Nowadays, we do not tune controllers “by hand”
- Instead: use state-of-the-art optimization algorithms
- Small optimization problems can be solved within micro-seconds and on embedded hardware



Embedded Optimization

Idea: always look a bit into the future.



Brain predicts and optimizes:
e.g. slow down **before** curve

Use repeated computation of optimal controls for feedback control!

Why Optimization + Control ?

Control objective:

Minimize time

Why Optimization + Control ?

Control objective:

Minimize fuel consumption/ safe energy

(subject to other tasks)

Why Optimization + Control ?

Control objective:

Maximize energy

Take Home Points

- Feedback control connects sensors and actuators

We tuned and simulated a closed loop system with 20 lines of code

Take Home Points

- With minor modifications we could use these 20 lines of code to tune and simulate an airplane autopilot

Take Home Points

- With minor modifications we could use these 20 lines of code to tune and simulate an airplane autopilot

Even simple controllers operate faster and more accurately than any human pilot

Take Home Points

- With minor modifications we could use these 20 lines of code to tune and simulate an airplane autopilot

Even simple controllers operate faster and more accurately than any human pilot

Just a few lines of code, no joke!

Take Home Points

Modern control methods use optimization algorithms