HUMAN-COMPUTER INTERACTION : HCI

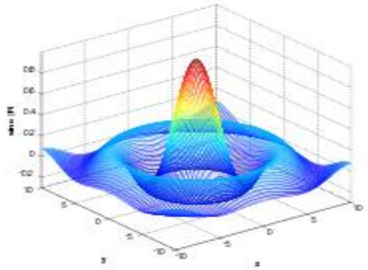
INSTRUCTOR: DR.MONTRI PHOTHISONOTHAI,

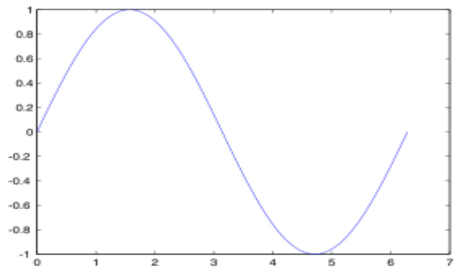
INTRODUCTION TO MATLAB

INTRODUCTION

Matlab (MATrix LABoratory) is an interactive software system for numerical computations and graphics.

It has a variety of graphical capabilities, and can be extended through programs written in its own programming language.





ABOUT MATLAB

MATLAB was created in the late 1970s by *Cleve Moler*, then chairman of the computer science department at the *University of New Mexico.*

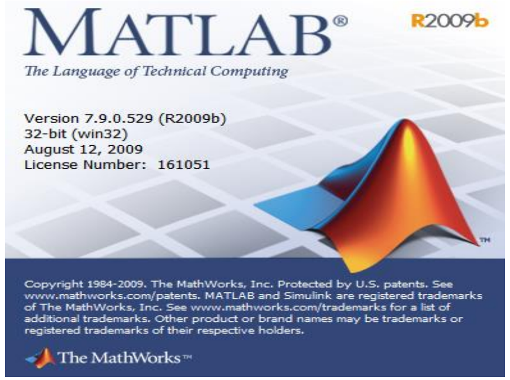
MATLAB written in *C* and founded by *MathWorks* in 1984 to continue its development

SYNTAX OF MATLAB

MATLAB, the application, is built around the MATLAB language.

The simplest way to execute MATLAB code is to type it in at the prompt, >> ,

in the Command Window, one of the elements of the MATLAB Desktop.



INSTALLATION OF MATLAB

MATLAB

- MATLAB command window main programming

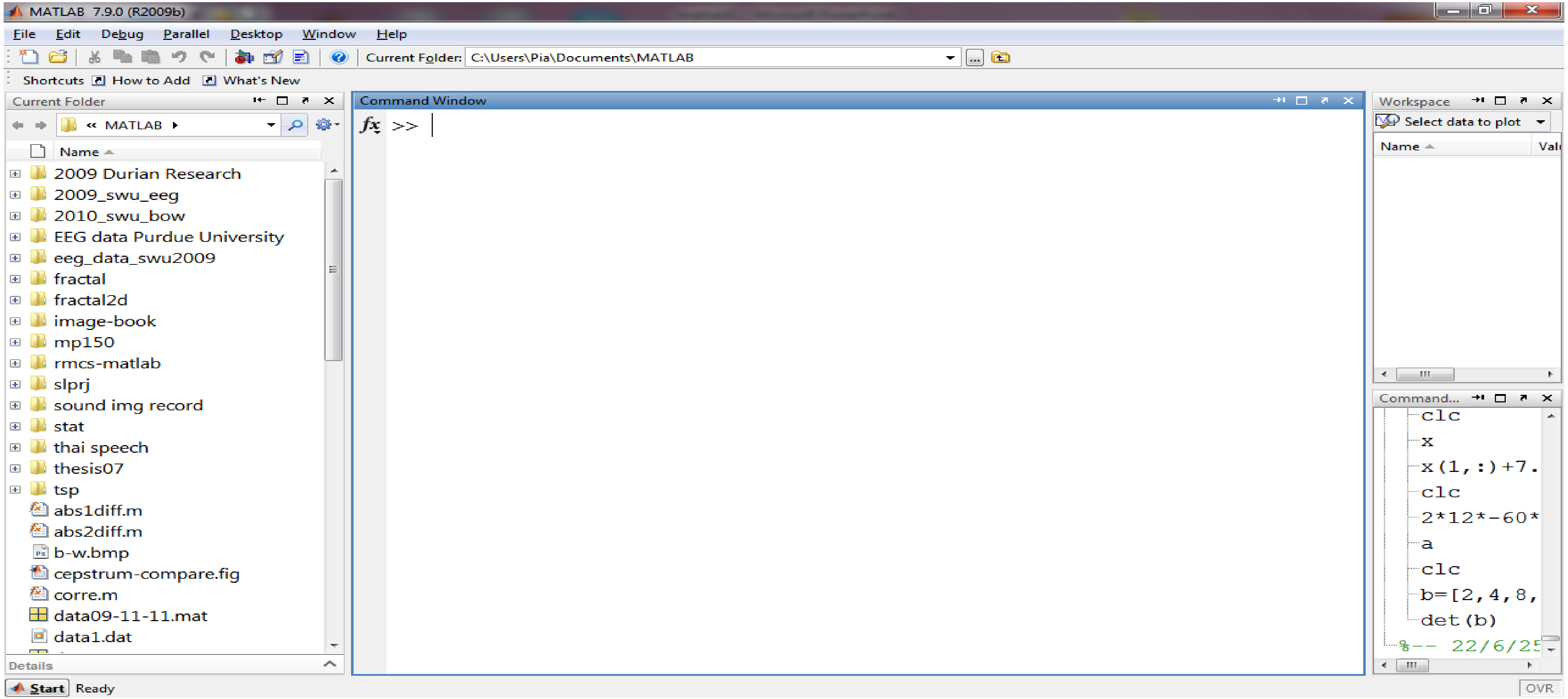
Toolboxes

- (Control, Biometrics, Neural Network; for examples)

Simulink

- Simulation-based experiments

MATLAB ENVIRONMENTS



CONTENTS

- ❑ Matlab as a calculator
- ❑ Powers, log, exponentials, trigonometry
- ❑ Defining matrices, NxN size
- ❑ Basic matrix operation

MATLAB AS A CALCULATOR

```
>> 1+1
>> 1 - - - 1
>> 1/2 + 3*4
>> 1\2 + 3*4
```

POWERS, LOG, EXPONENTIALS, TRIGONOMETRY

```
>> 2^3
>> log(2)    // returns the natural logarithm
>> exp(e)
>> pi
>> sin(pi)
>> cos(pi/2)
>> tan(pi/4)
>> help pi
```

ENTERING VECTORS AND MATRICES; BUILT-IN VARIABLES AND FUNCTIONS; HELP

```
>> a = 2
a = 2
```

```
>> x = [1;2;3]
x = 1 2 3
```

```
>> A = [1 2 3;4 5 6;7 8 0]
A = 1 2 3 4 5 6 7 8 0
```

EXAMPLE

$$A = \begin{bmatrix} 8 & 3 \\ 2 & -5 \\ -9 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 7 & -6 \\ -1 & 2 \\ 3 & 0 \end{bmatrix}$$

$$A - B = A + -1 * B$$

$$= \begin{bmatrix} 8 & 3 \\ 2 & -5 \\ -9 & 1 \end{bmatrix} + -1 * \begin{bmatrix} 7 & -6 \\ -1 & 2 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 3 \\ 2 & -5 \\ -9 & 1 \end{bmatrix} + \begin{bmatrix} -1*7 & -1*(-6) \\ -1*(-1) & -1*2 \\ -1*3 & -1*0 \end{bmatrix}$$

$$= \begin{bmatrix} 8 & 3 \\ 2 & -5 \\ -9 & 1 \end{bmatrix} + \begin{bmatrix} -7 & 6 \\ 1 & -2 \\ -3 & 0 \end{bmatrix} = \begin{bmatrix} 8+-7 & 3+6 \\ 2+1 & -5+-2 \\ -9+-3 & 1+0 \end{bmatrix} = \begin{bmatrix} 1 & 9 \\ 3 & -7 \\ -12 & 1 \end{bmatrix}$$

$$\mathbf{D} \quad * \quad \mathbf{A} \quad = \quad \mathbf{C}$$

$$\begin{bmatrix} 2 & 1 & 3 \\ -2 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 1 \\ 3 & 2 \\ -2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 10 \\ 0 & 4 \end{bmatrix}$$

SUMMATION

1+2+3+…+10

>> sum(1:10)

$$y = \sum_{n=-5}^{5} (1 + n^2)$$

$$y = \sum_{n=1}^{1000} \frac{1}{2n}$$

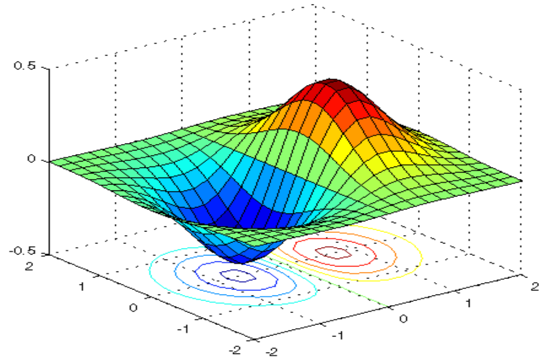
CONTENTS

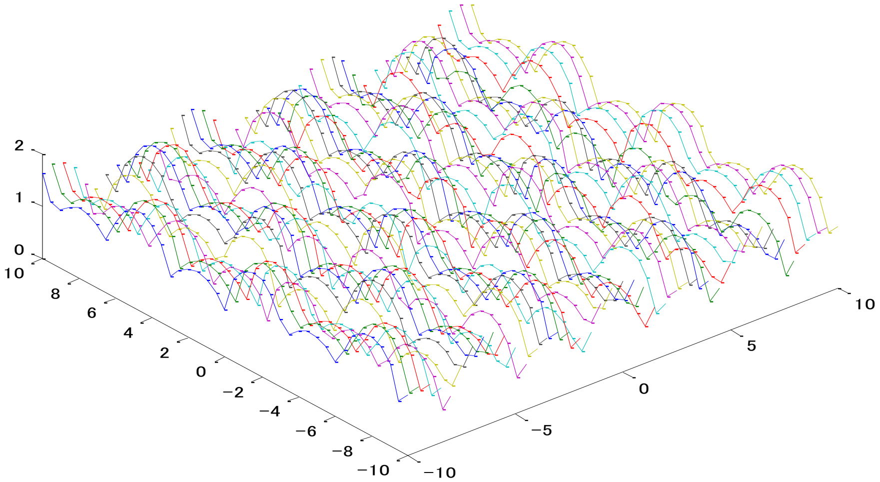
Graphics and visualization

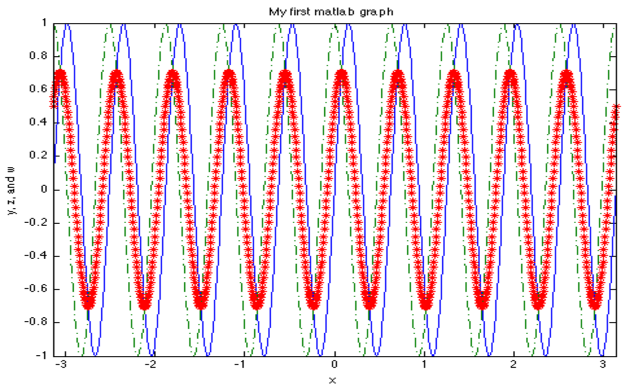
Basic plot in 2D and 3D

Figure edit

Function and scripts







EXAMPLE

2D plot

(*one dependent variable*, *one independent variable*)

Command

>> plot(x, y)

>> bar(x, y)

2D LINE PLOT

Step size is a key to plot the graph

```
>> x = 0:.01:20;

>> y = x.^2;

>> plot(t,y);
```

EXERCISES

Step size = 0.1, interval *x* = -10 to 10

$$y = x^3$$

```
y = x.^3
```

$$y = \frac{1}{x}$$

```
y = 1./x
```

$$y = \frac{\sin x}{x}$$

```
y = sin(x)./x
```

$$y = e^{-0.05t} \sin 5t$$

```
y = exp(-0.05.*t).*sin(5.*t)
```

2D LINE PLOT

```
>> t = -10:0.1:10;
>> alpha =.055;
```

$$y = e^{-\alpha t} \sin 5t$$

```
>> y = exp(-alpha*t).*sin(.5*t);
```

```
>> plot(t,y);
```

EXAMPLE

3D plot

(*one dependent variable*, *two independent variables*)

Command

>> plot3(x, y, z)

>> mesh(z)

>> surf(z)

3D SURFACE PLOT

```
>> [x,y] = meshgrid([-2:.2:2]);
```

$$z = xe^{-x^2-y^2}$$

```
>> z = x.*exp(-x.^2-y.^2);
```

```
>> surf(z);
```

EXERCISES

Step size = 0.1, interval *x* = -5 to 5, *y* = -5 to 5

$$z = x^4 - y^4$$

$$z = \frac{\sin(x + y)}{(x + y)}$$

$$z = \frac{\sin x}{1 + \cos y}$$

INTRODUCTION TO SYSTEM MODELLING

Project (HCI) → Idea(s) → Implement → Applications

MATLAB

C, Java, Python …

DETERMINISTIC AND STOCHASTIC PROCESSES

Simulation modelling

- **Deterministic Model is a system in which no randomness is involved in the development of future states of the system. A deterministic model will thus always produce the same output from a given starting condition or initial state.**

- **Stochastic Model → random → probability modeling → uniform or normal**
  - Stock exchange, Lottery, Rolling dice, Tossing coin, …

PROBABILITY DISTRIBUTION

Here's a probability distribution for one roll of a six-sided die:

Take trials close to infinity

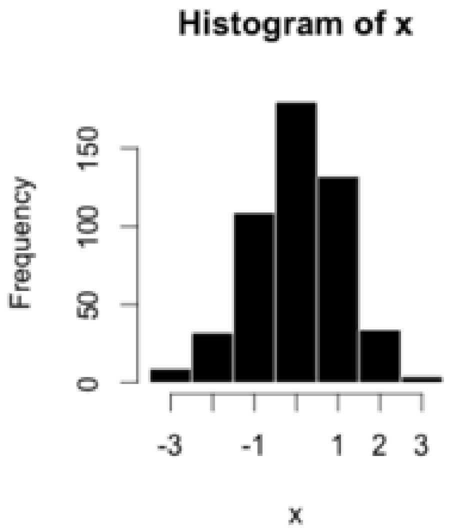
Number	P(Number)
1	1/6 = .167
2	1/6 = .167
3	1/6 = .167
4	1/6 = .167
5	1/6 = .167
6	1/6 = .167

PROBABILITY HISTOGRAM

A histogram is a graphical representation of the distribution of data. It is an estimate of the probability distribution of a continuous variable (quantitative variable) and was first introduced by Karl Pearson

The probability histogram is very similar to a relative frequency histogram, but the vertical scale shows probabilities.

Bin number versus Frequency (Occurrences)



MATLAB EXPERIMENT

```
x1 = rand(1,10);
x2 = rand(1,100);
x3 = rand(1,1000);

subplot(3,1,1); hist(x1);
subplot(3,1,2); hist(x2);
subplot(3,1,3); hist(x3);


x4 = randi(6,1,1000);
a = hist(x4);
bar(a/sum(a))
```

MATLAB EXPERIMENT

```
x1 = randn(1,10);
x2 = randn(1,100);
x3 = randn(1,1000);

subplot(3,1,1); hist(x1);
subplot(3,1,2); hist(x2);
subplot(3,1,3); hist(x3);
```

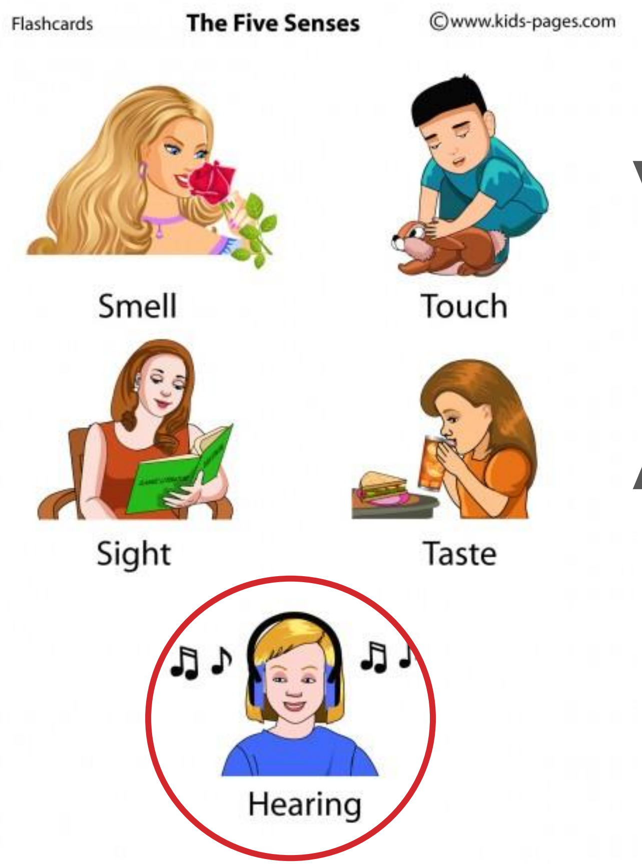
PROBABILISTIC MODELS

- Bayesian Decision Theory
- Prior Probability

AUDIO SIGNAL PROCESSING BY MATLAB

AUDIO, SPEECH, VOICE

It is one of the traditional five senses : Hearing is one of the most important senses that humans have available.



Sensors (microphone, soundcard)

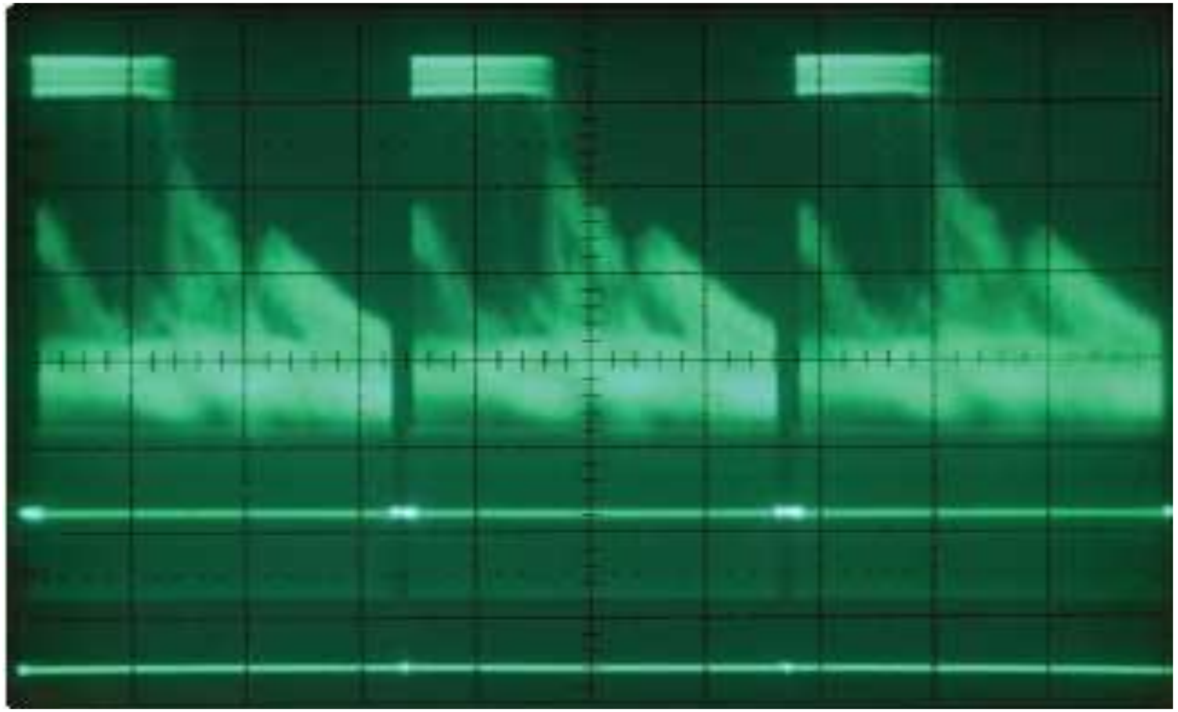
Digital format (wav, mp3, pcm)

Process (algorithms, software)

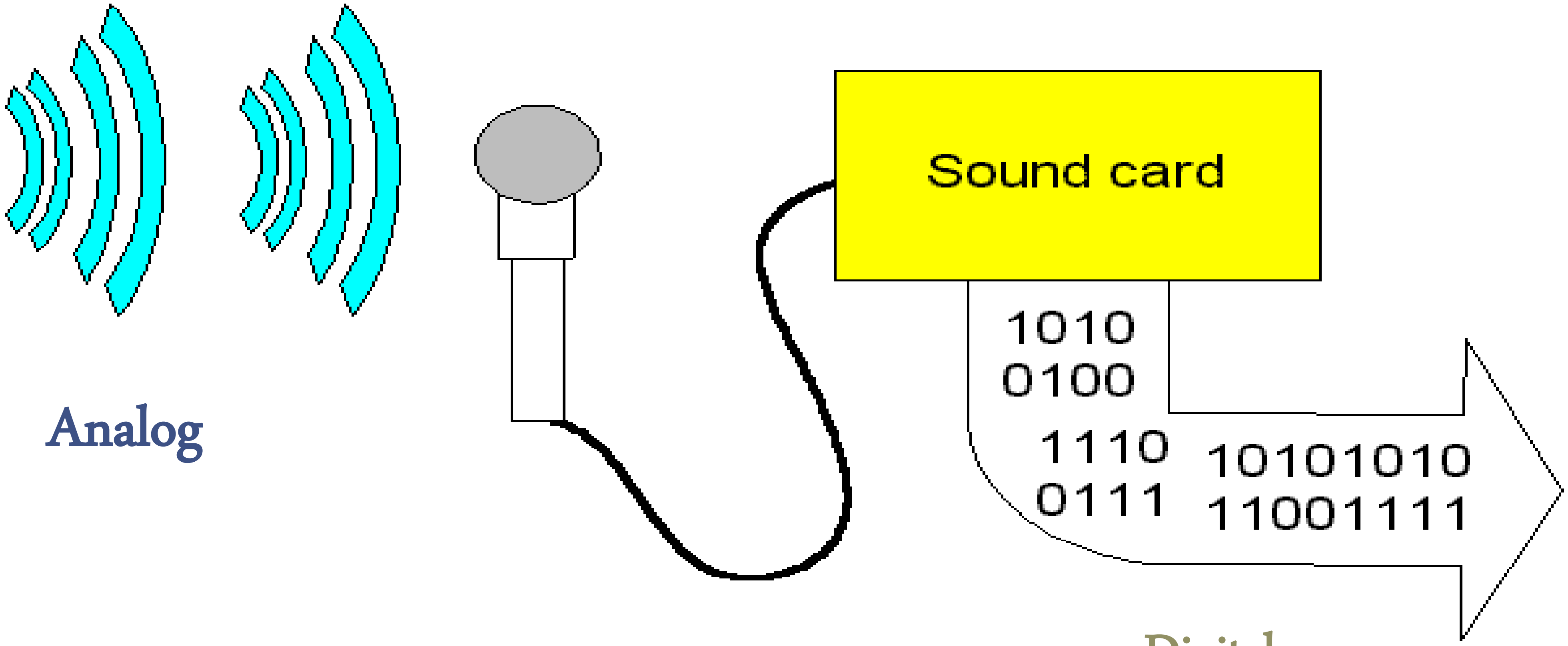
CONTENTS

- ❑ Analog-to-Digital Conversion
- ❑ Sampling theory
- ❑ Time domain representation
- ❑ Fourier theory
- ❑ Frequency domain representation
- ❑ Feature extraction algorithms

REAL WAVEFORM OF ANALOG SIGNALS

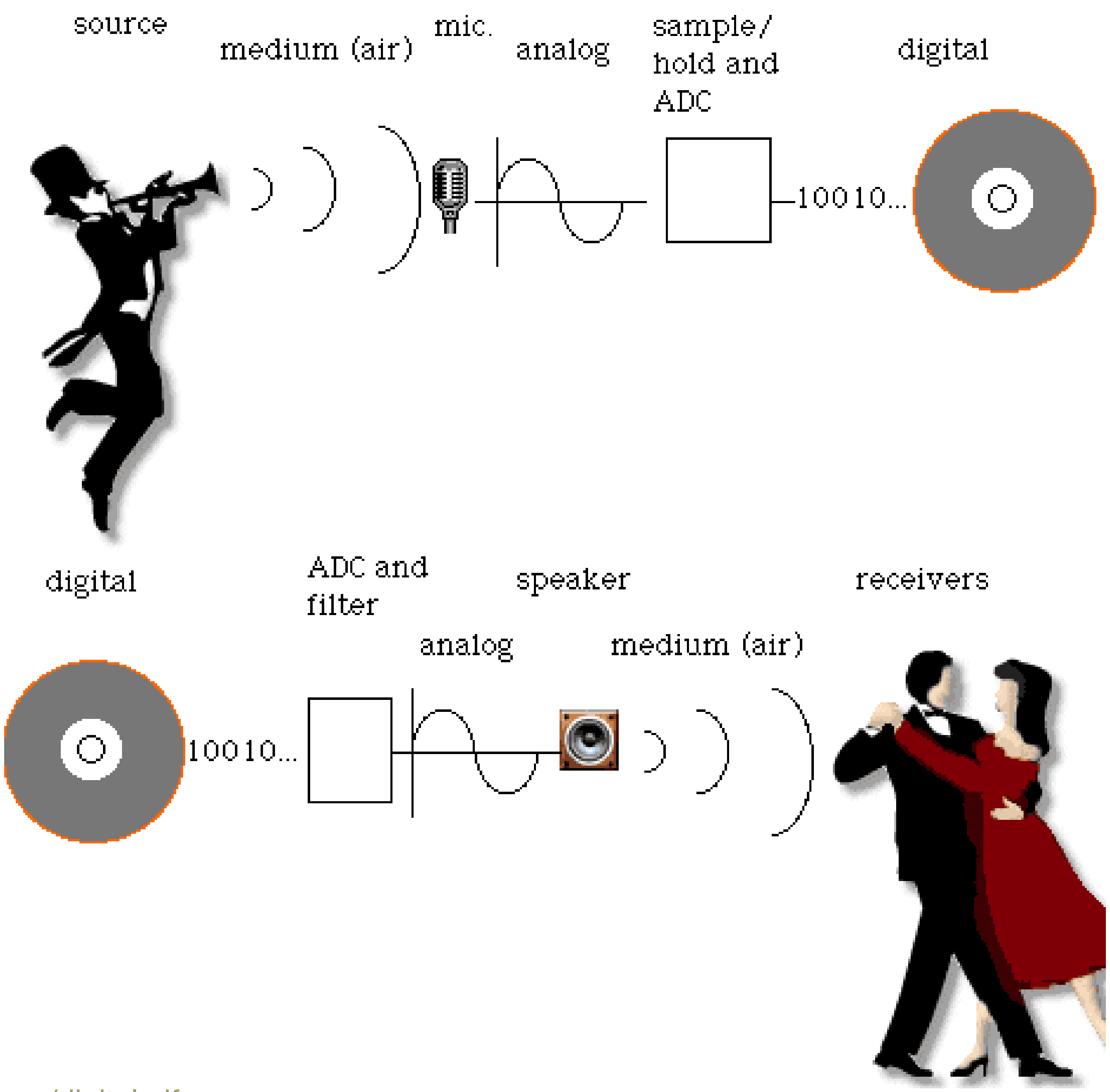


ANALOG VS. DIGITAL



Analog

Digital



http://www.atpm.com/6.02/images/digital.gif

ANALOG TO DIGITAL CONVERSION

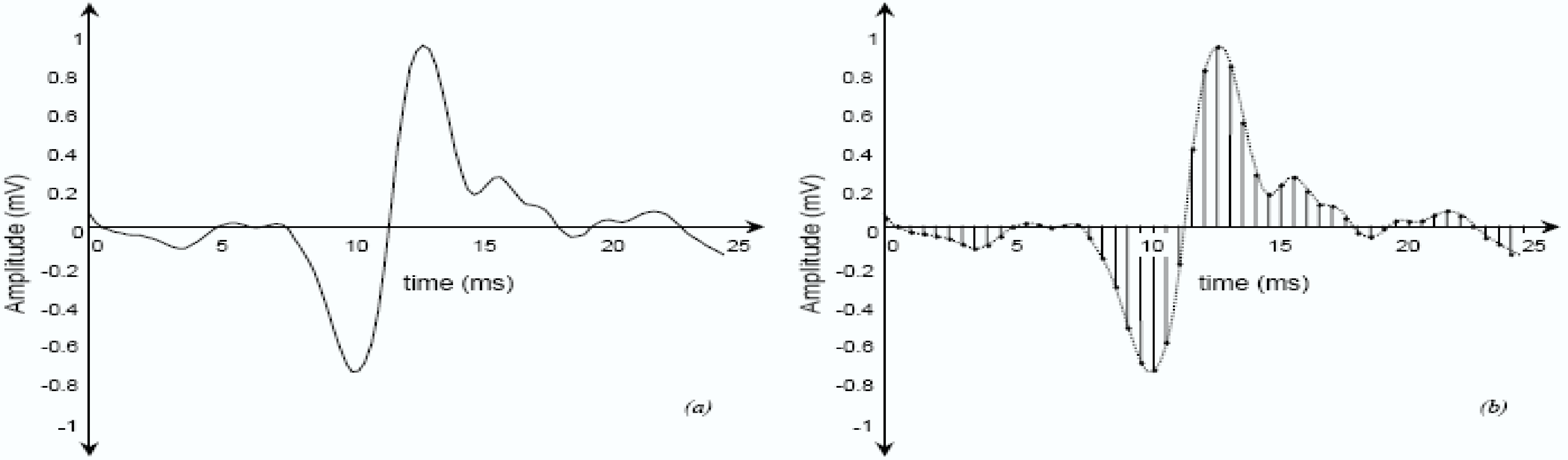
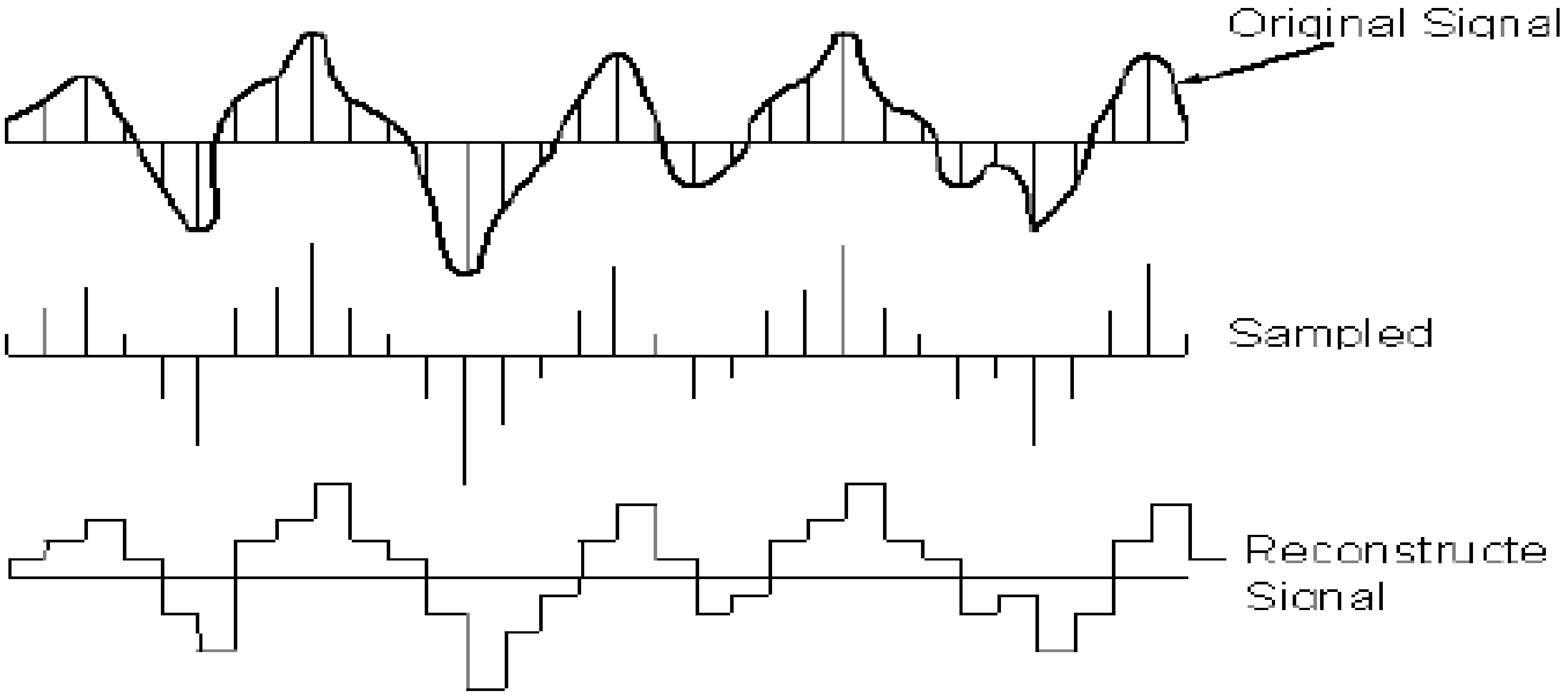


Figure 1: a) A typical analog EMG signal detected by the DE-2.1 electrode. (b) The digital sequence resulting from sampling the signal in (a), at 2 kHz (every 0.5 ms).

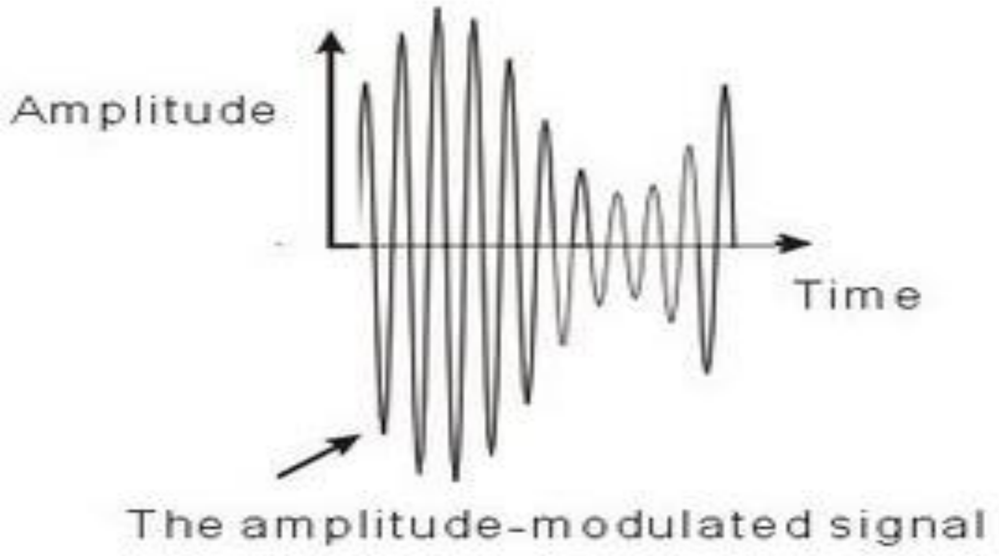
SAMPLING

•The sampling is an analog, not digital, process and is accomplished with a "sample and hold" circuit. The output of this circuit is a sequence of voltage levels that are fed into an analog-to-digital converter (ADC).



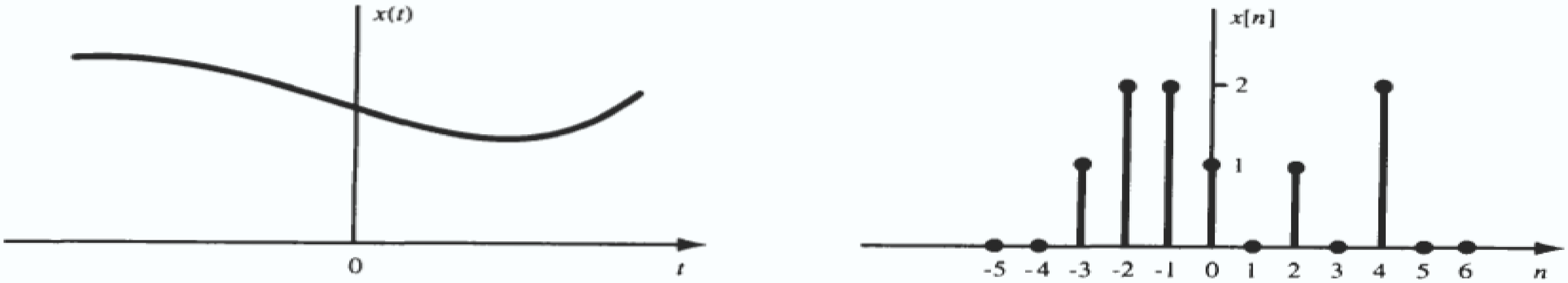
SIGNAL IN TIME DOMAIN

•Time domain is a term used to describe the analysis of mathematical functions, or physical signals, with respect to time.



CONTINUOUS VS. DISCRETE-TIME SIGNALS

A signal $x(t)$ is a *continuous-time* signal if t is a continuous variable. If t is a discrete variable, that is, $x(t)$ is defined at discrete times, then $x(t)$ is a *discrete-time* signal. Since a discrete-time signal is defined at discrete times, a discrete-time signal is often identified as a *sequence* of numbers, denoted by $\{x_n\}$ or $x[n]$, where n = integer. Illustrations of a continuous-time signal $x(t)$ and of a discrete-time signal $x[n]$ are shown in Fig. 1-1.

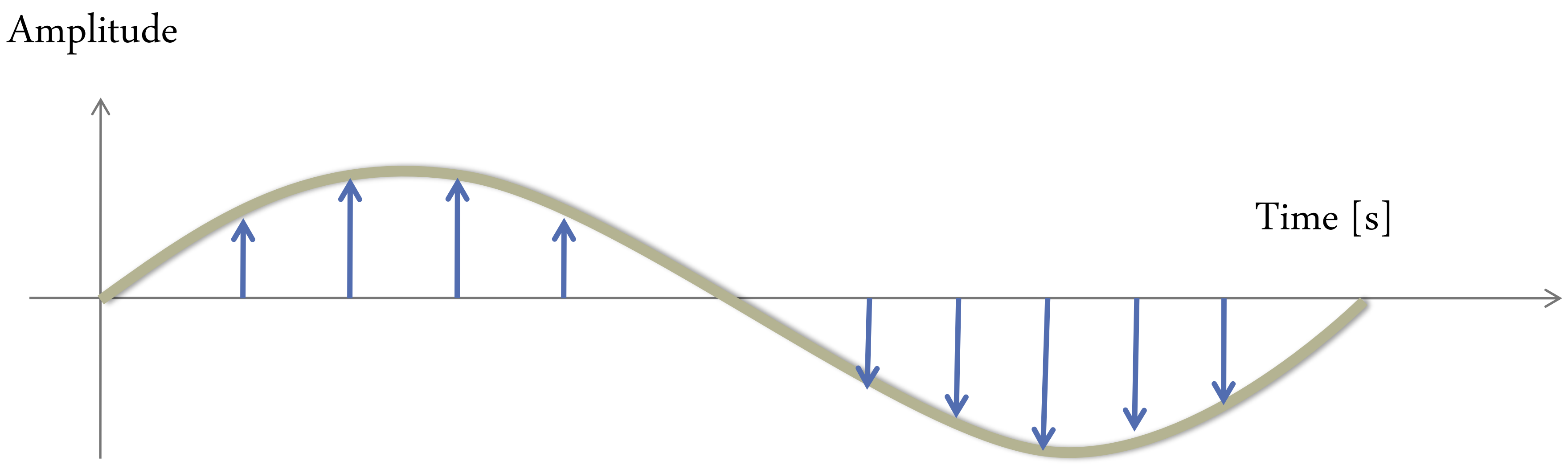


(*a*) continuous-time and (*b*) discrete-time signals.

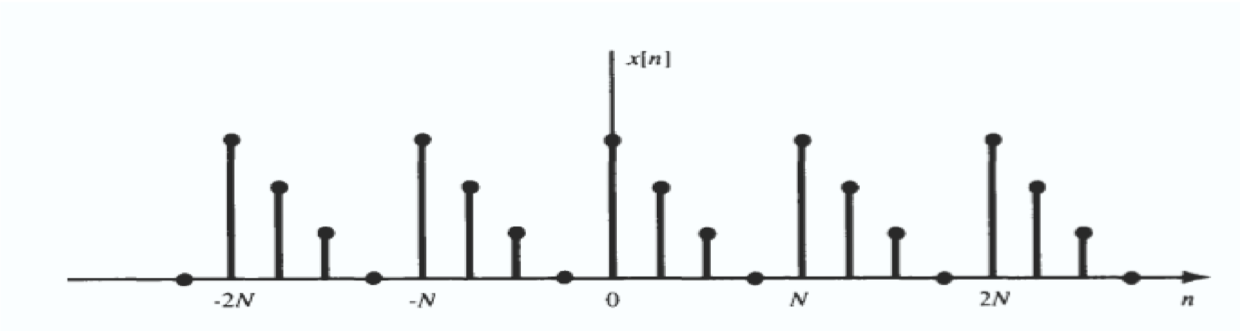
SAMPLING RATE

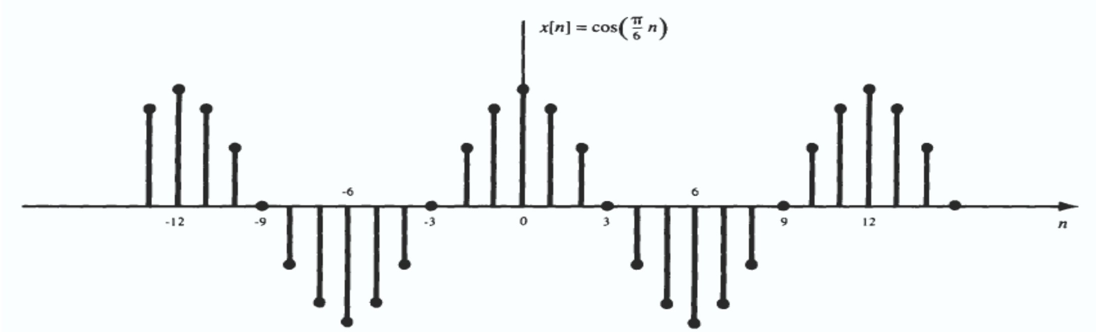
Suitable sampling rate

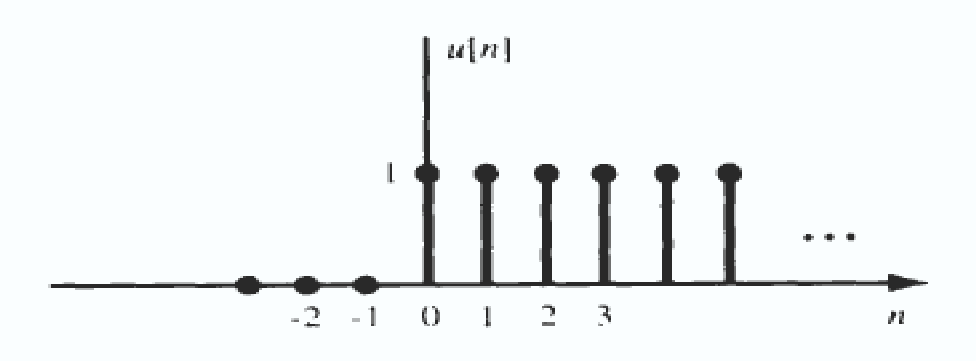
Or called sampling frequency [Hz]

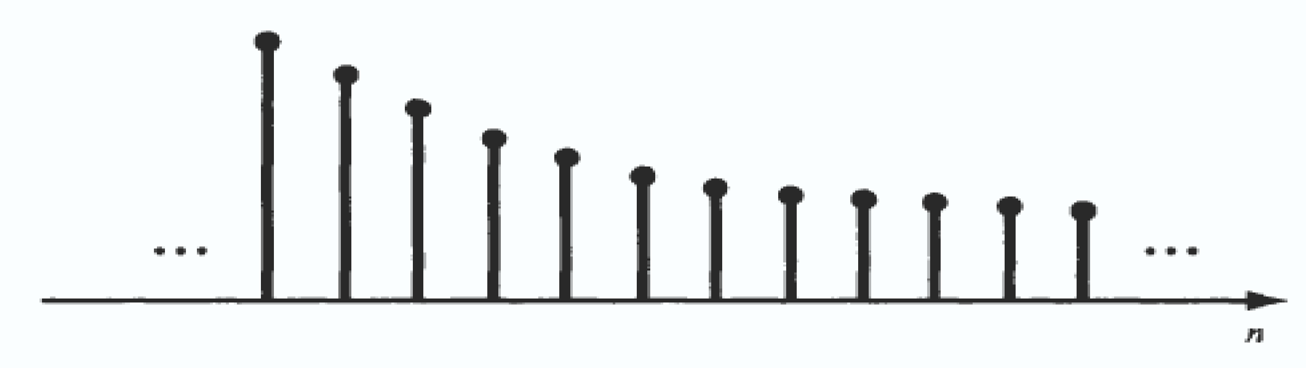


SAMPLED SIGNALS OR DISCRETE-TIME SIGNALS

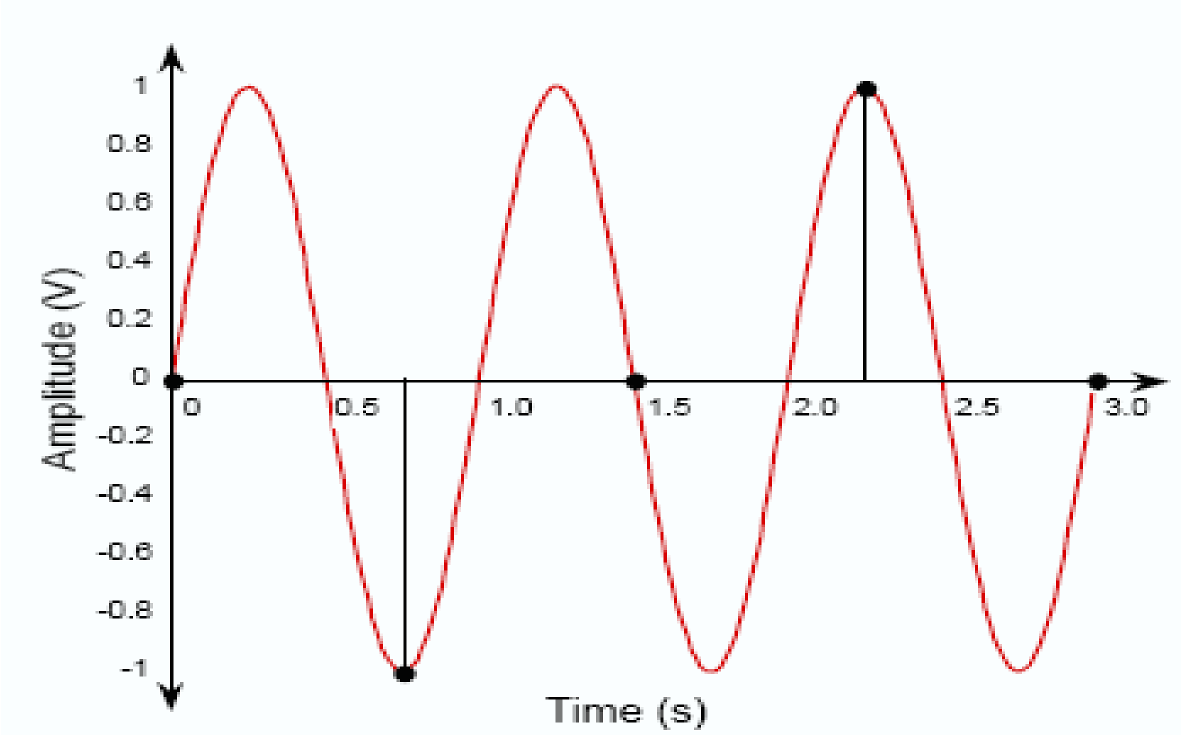


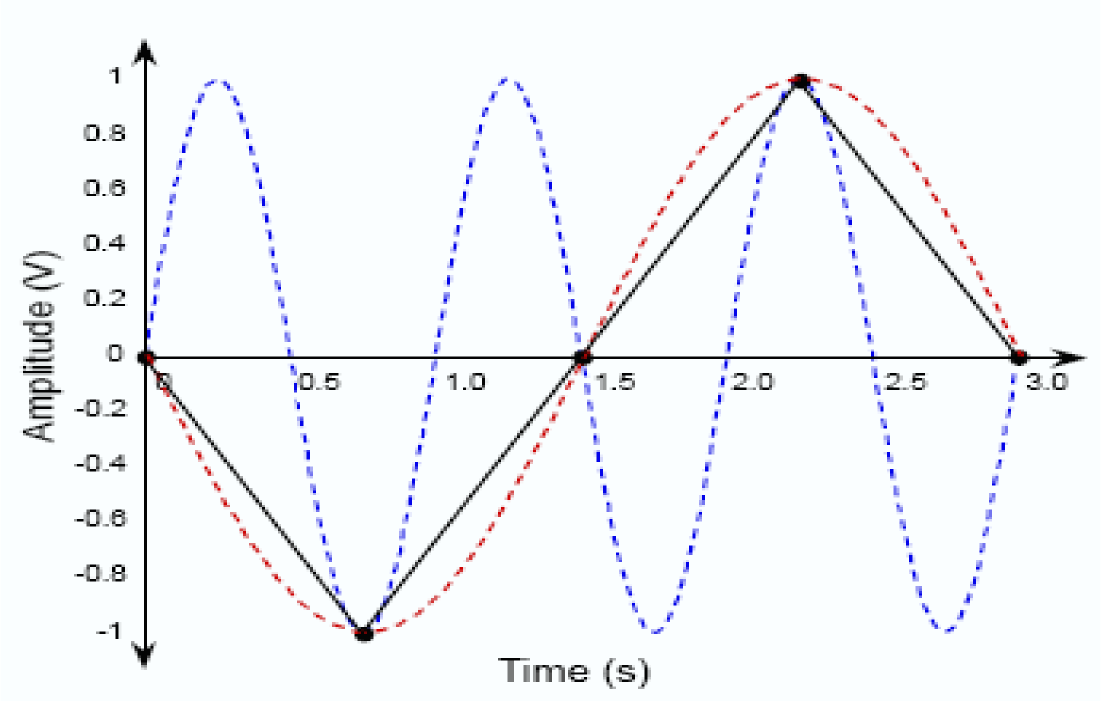






PROBLEM OF SAMPLING





PERFECT SAMPLING RATE

How do we know the extract sampling rate?

Harry Nyquist

it is sampled *at no less than twice its frequency.*

It is "Nyquist rate" that can be defined as:

$$f_s \geq 2 f_n$$

EXAMPLES

Original source generates Amplitude 5 V with 10 Hz

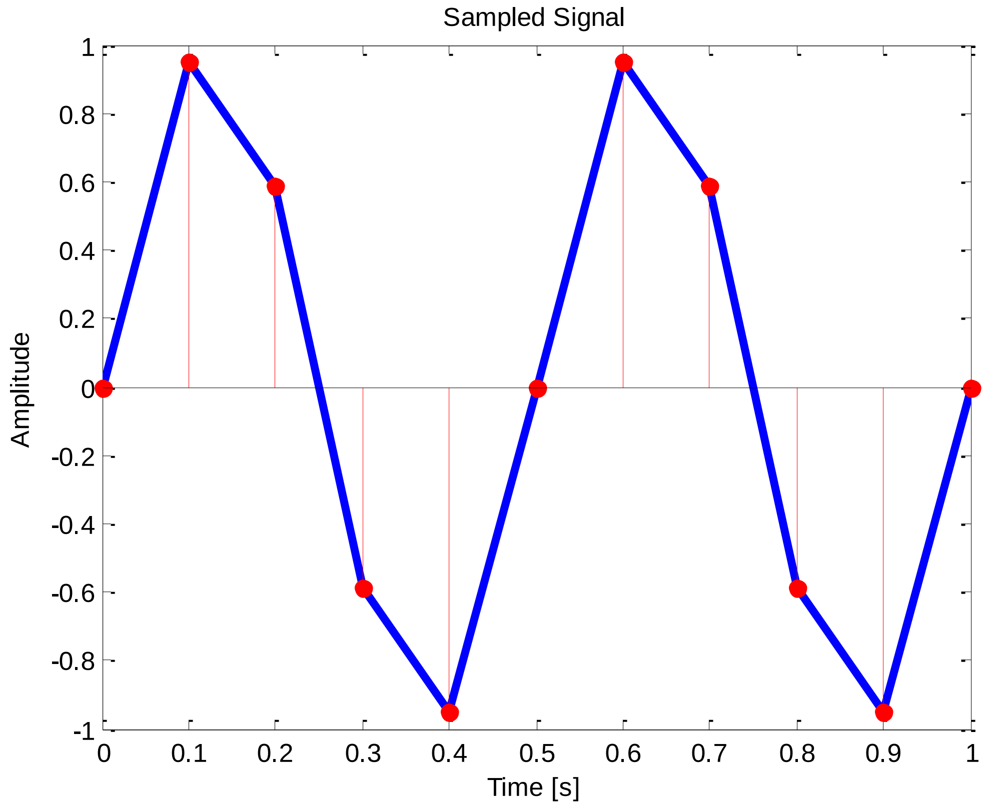
- Show the output waveform in one circle.
- The proper rate to sample this data is _______ Hz ?

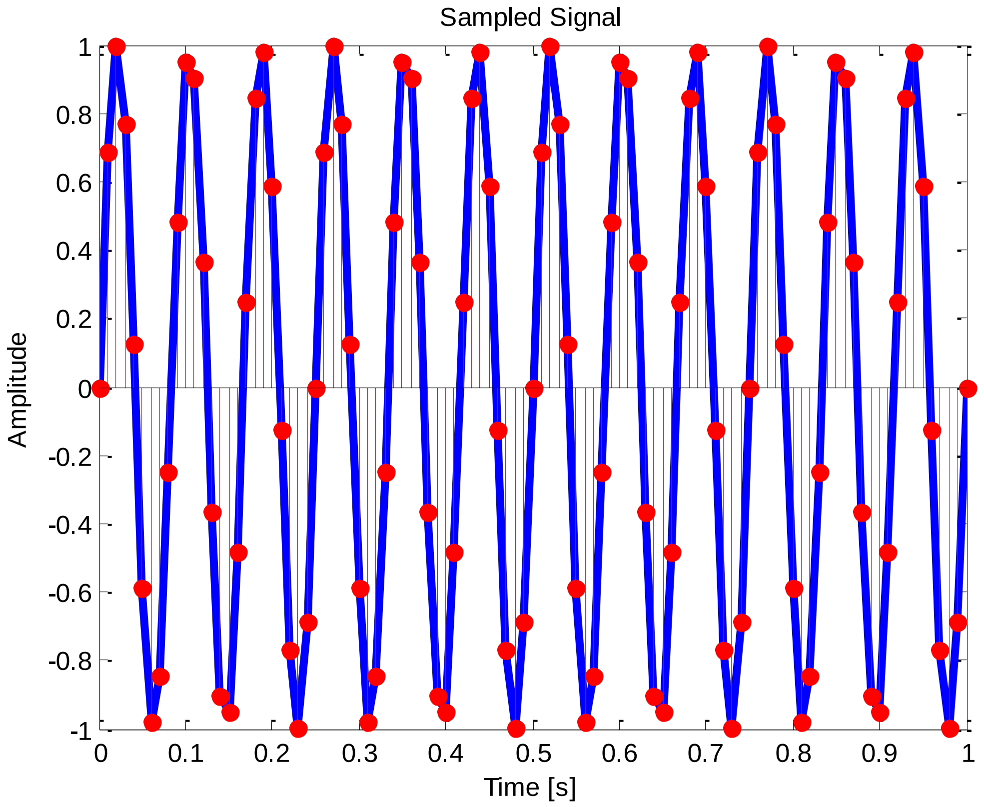
MATLAB EXPERIMENTS

Code [Sampling1]

Fs = 10 Hz, 100 Hz

Fo = 12 Hz, 12 Hz





APPLICATIONS

it is necessary to capture audio covering the entire 20–20,000 Hz range of human hearing

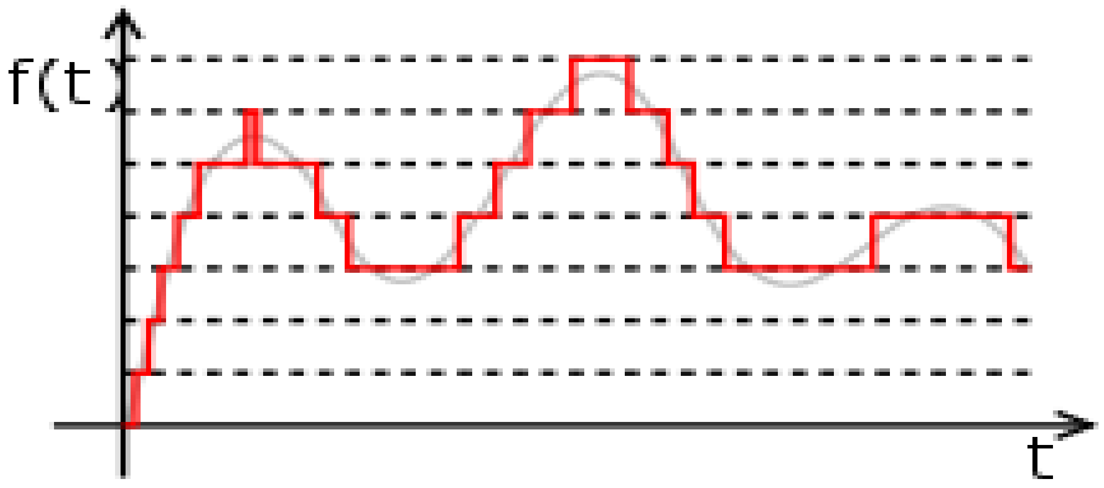
8,000 Hz: Telephone and encrypted walkie-talkie, wireless intercom and wireless microphone transmission

11,025 Hz: One quarter the sampling rate of audio CDs; used for lower-quality PCM, MPEG audio

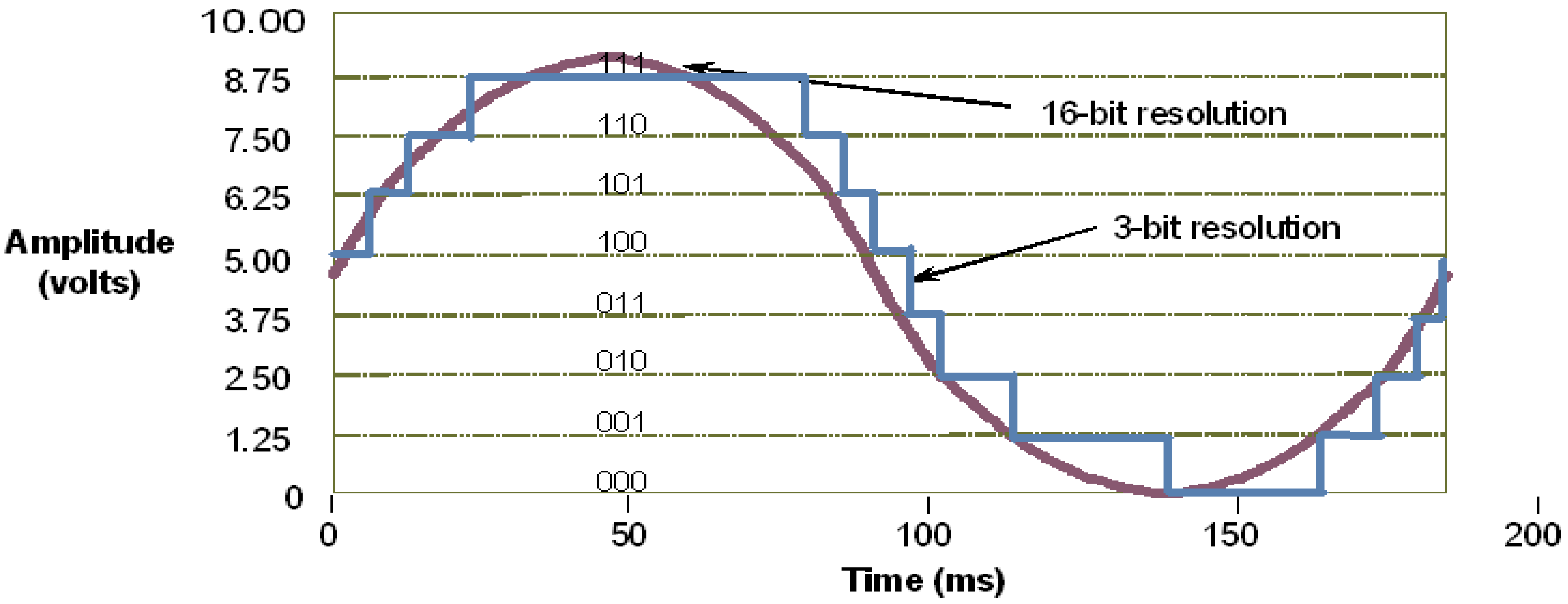
44,100 Hz: Audio CD, also most commonly used with MPEG-1 audio (VCD, SVCD, MP3). Originally chosen by Sony

QUANTIZATION

the process of approximating ("mapping") a continuous range of values by a relatively small set of discrete symbols or integer values.



RESOLUTION (BIT)



EXAMPLE

Assume that $f(t) = \sin t$ where t = [0…1] step size by 0.1

The quantization process for this signal is:

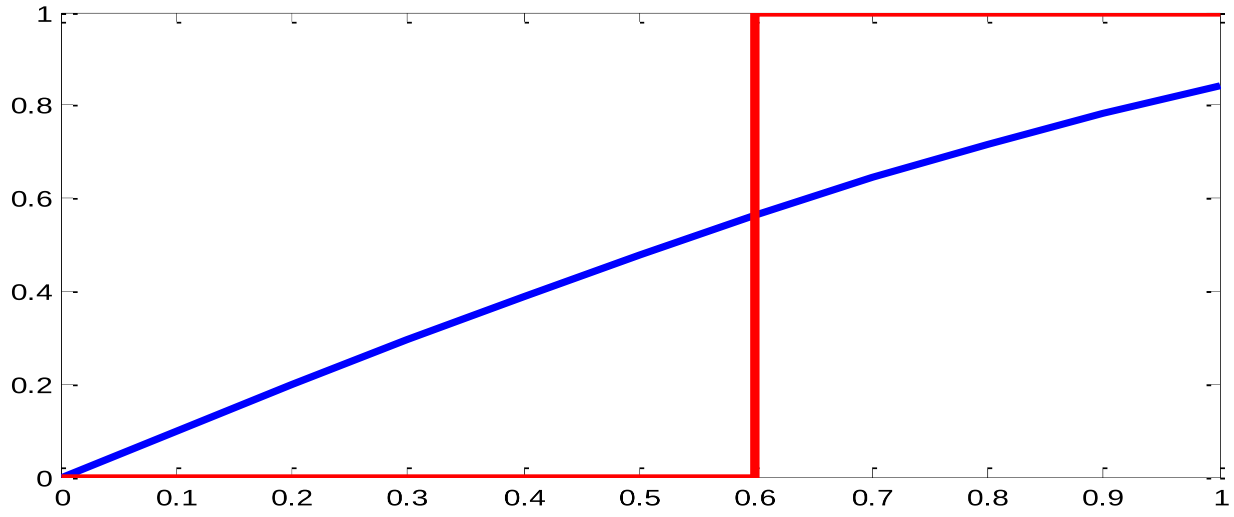
- 1-bit resolution (2 levels)
- 2-bit resolution (4 levels)
- 3-bit resolution (8 levels)
- 4-bit resolution (16 levels)

EXAMPLE (1-BIT RESOLUTION)

t	f(t)	1bit
0	0	0
0.1	0.0998	0
0.2	0.1987	0
0.3	0.2955	0
0.4	0.3894	0
0.5	0.4794	0
0.6	0.5646	1
0.7	0.6442	1
0.8	0.7174	1
0.9	0.7833	1
1	0.8415	1

$$Q = \frac{Range}{Level}$$

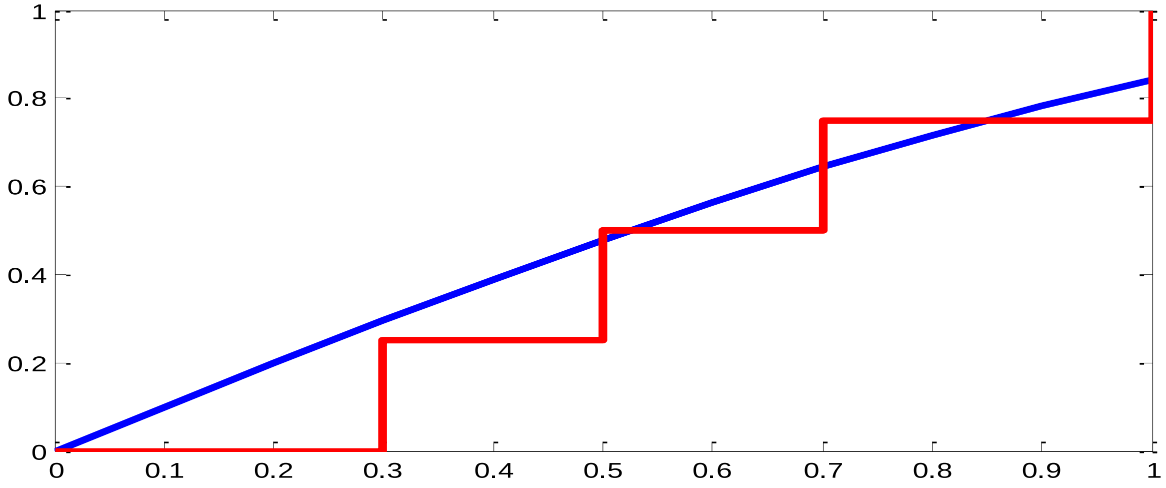
$$Q = \frac{1}{2}$$



EXAMPLE (2-BIT RESOLUTION)

t	f(t)	2bit
0	0	0
0.1	0.0998	0
0.2	0.1987	0
0.3	0.2955	0.25
0.4	0.3894	0.25
0.5	0.4794	0.5
0.6	0.5646	0.5
0.7	0.6442	0.75
0.8	0.7174	0.75
0.9	0.7833	0.75
1	0.8415	1

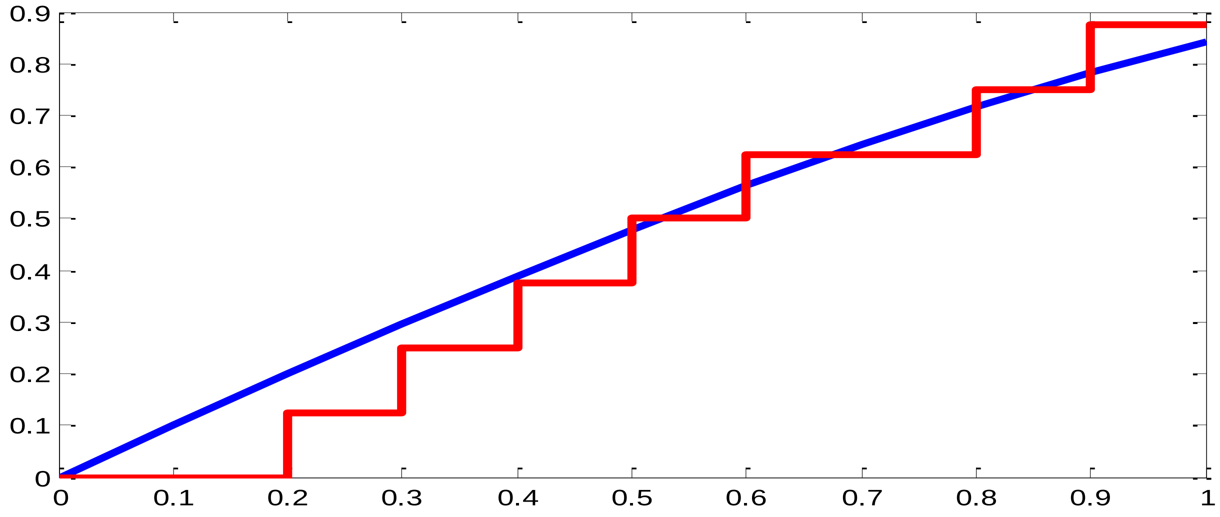
$$Q = \frac{1}{4} = 0.25$$



EXAMPLE (3-BIT RESOLUTION)

t	f(t)	3bit
0	0	0
0.1	0.0998	0
0.2	0.1987	0.125
0.3	0.2955	0.25
0.4	0.3894	0.375
0.5	0.4794	0.5
0.6	0.5646	0.625
0.7	0.6442	0.625
0.8	0.7174	0.75
0.9	0.7833	0.875
1	0.8415	0.875

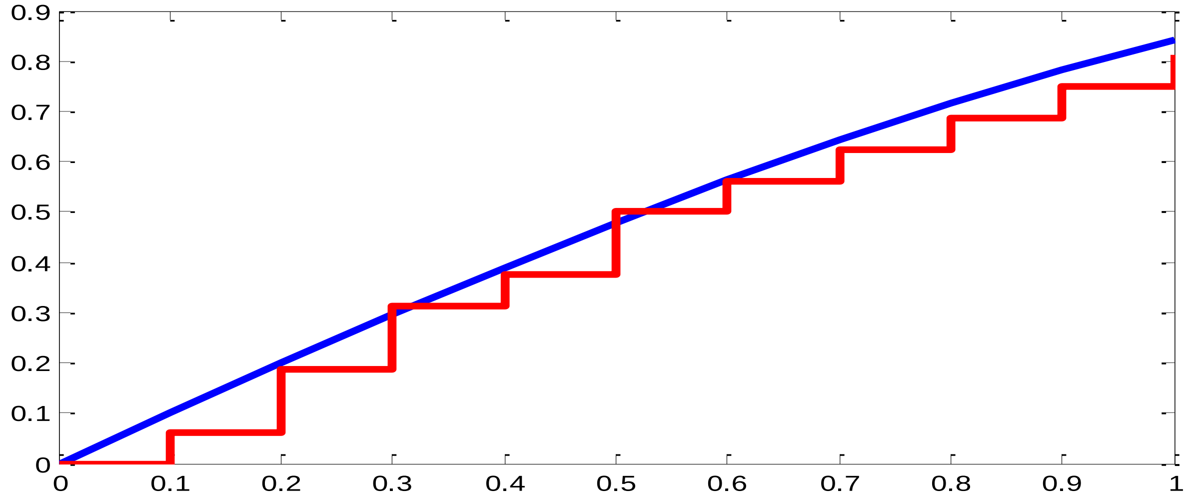
$$Q = \frac{1}{8} = 0.125$$



EXAMPLE (4-BIT RESOLUTION)

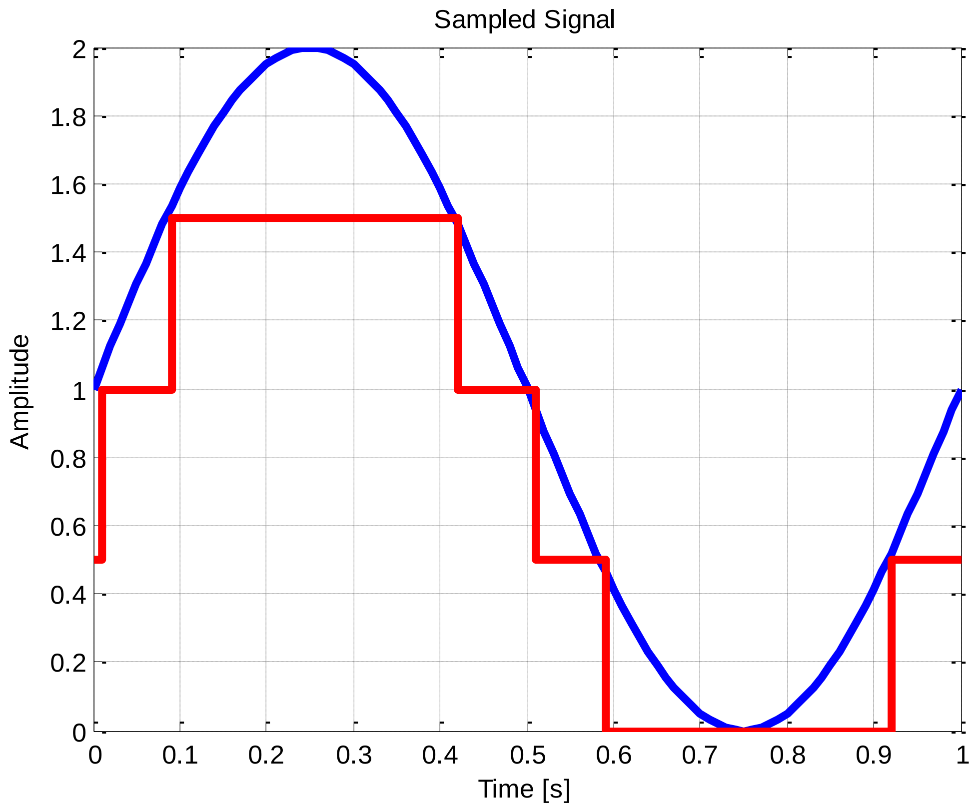
t	f(t)	4bit
0	0	0
0.1	0.0998	0.0625
0.2	0.1987	0.1875
0.3	0.2955	0.3125
0.4	0.3894	0.375
0.5	0.4794	0.5
0.6	0.5646	0.5625
0.7	0.6442	0.625
0.8	0.7174	0.6875
0.9	0.7833	0.75
1	0.8415	0.8125

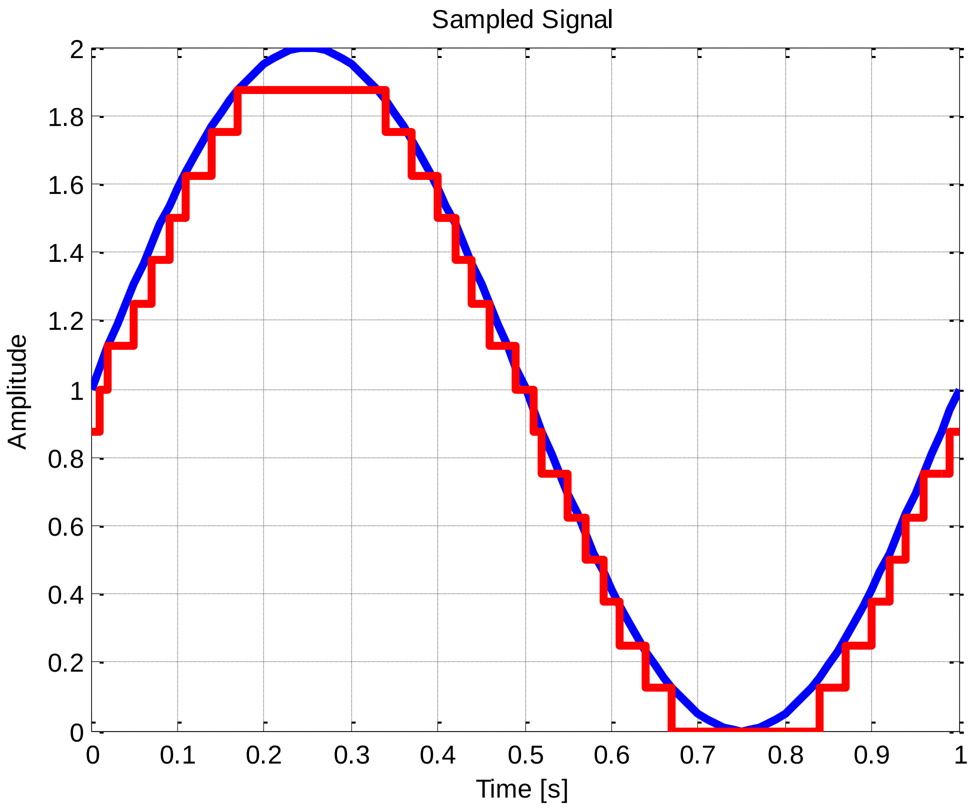
$$Q = \frac{1}{16} = 0.0625$$



MATLAB EXPERIMENTS

Code [Sampling2]





CONTENTS

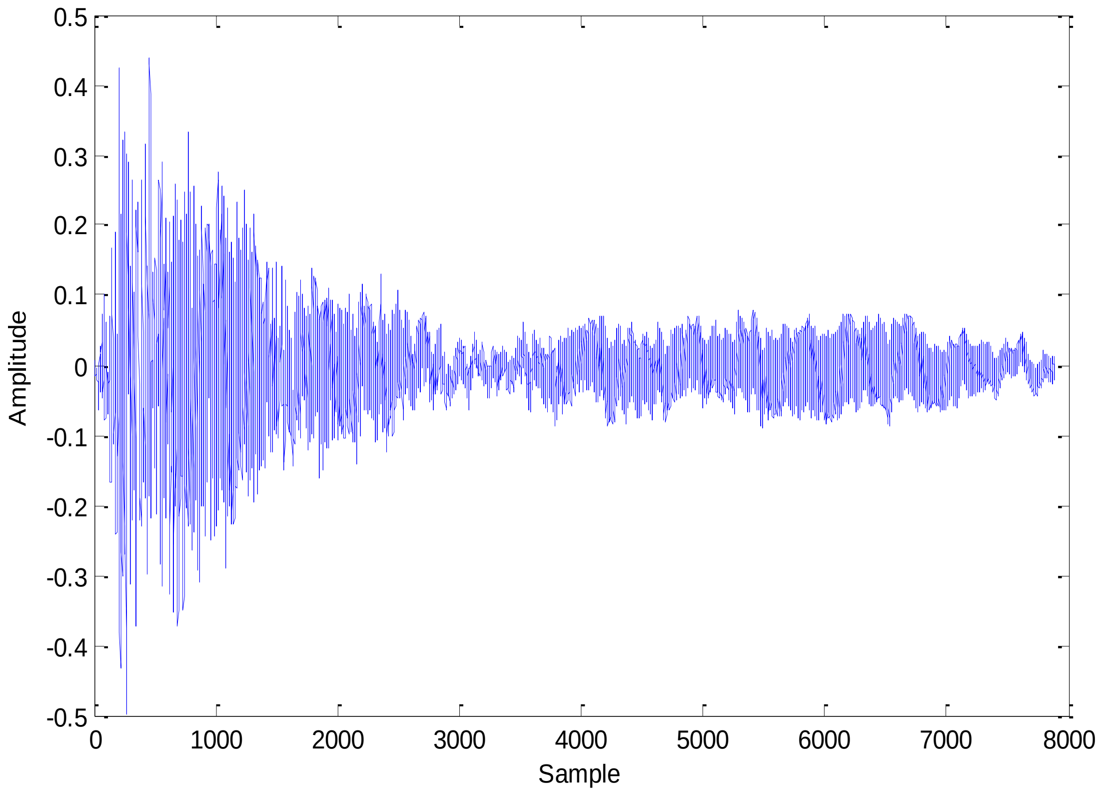
PIANO EXPERIMENT

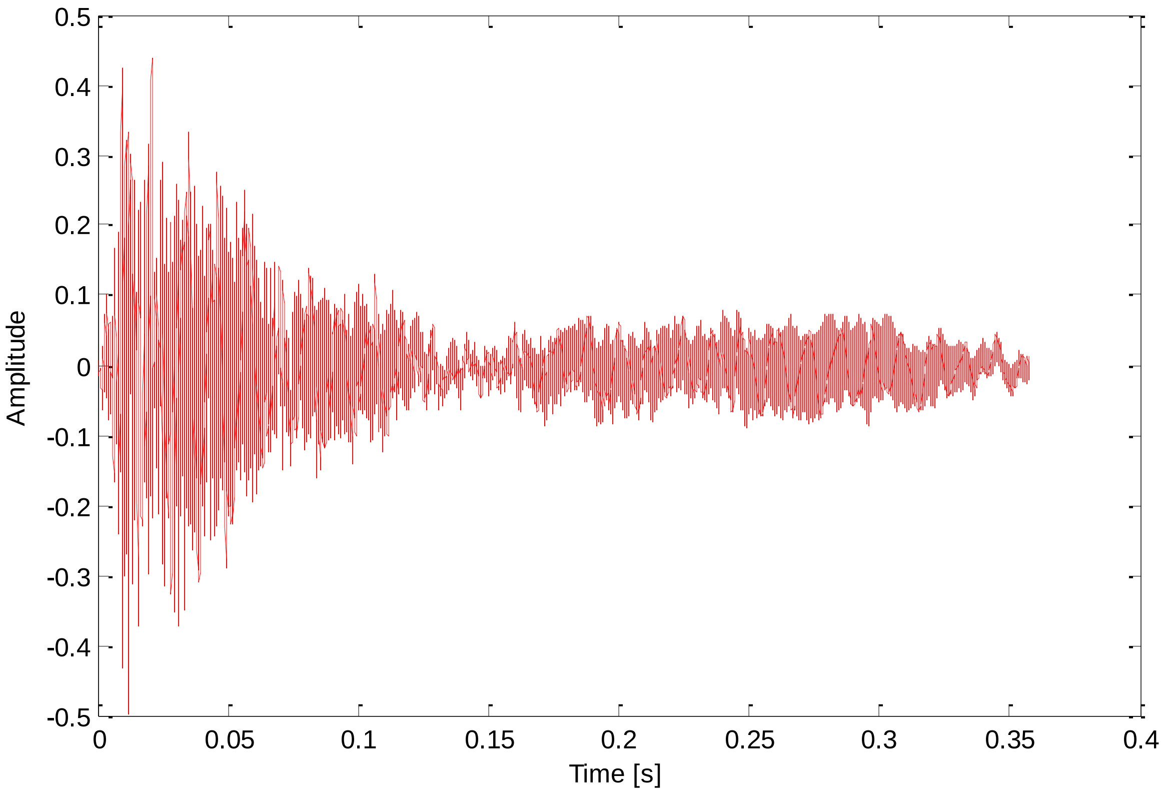
- **Load “Piano.zip”**
- **Extract to the working directory of MATLAB**
- **Double click the “Piano” folder**
- **Enter command**

```
>> [x1, fs] = wavread('high.wav');
>> [x2, fs] = wavread('mid.wav');
>> [x3, fs] = wavread('low.wav');
>> sound(x1, fs)
>> wavplay(x1, fs)
```

TIME DOMAIN

```
N = length(x1);
t = 0 : 1/fs : N/fs-1/fs;
plot(t, x1, 'r');
```





NORMALIZATION

In the simplest cases, normalization means adjusting values measured on different scales to a notionally common scale, often prior to averaging.

Normalization is needed for the preprocessing part of the system.

Procedures

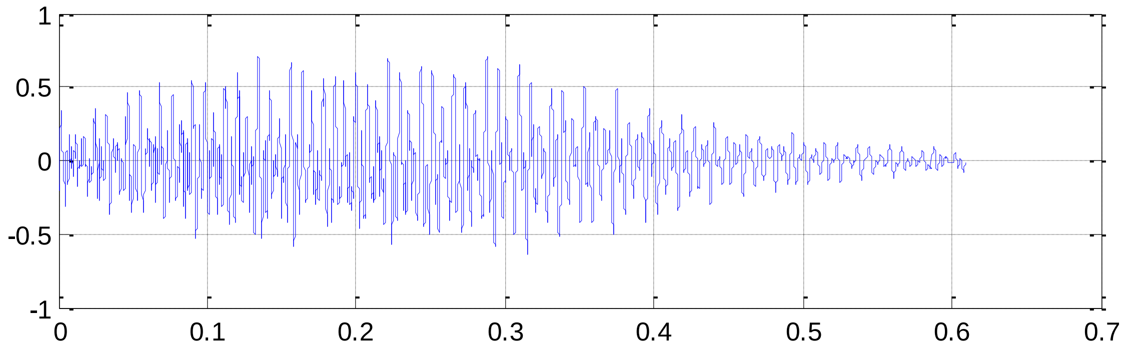
- Zero mean
- Unit variance

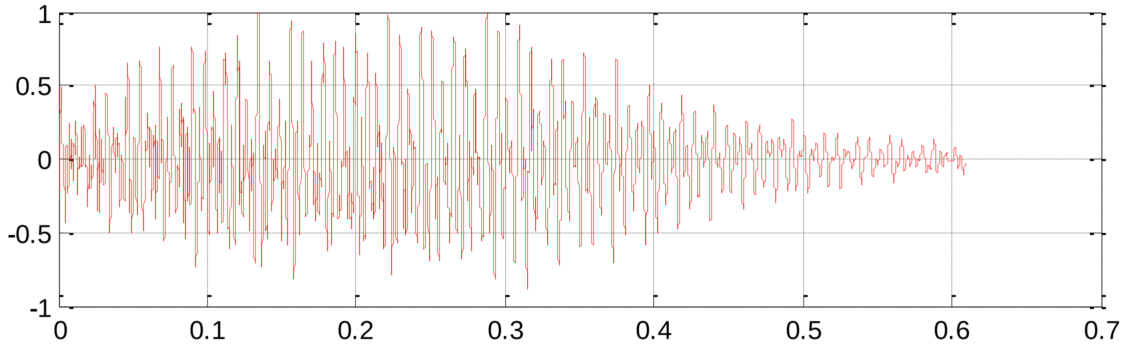
MATLAB EXERIMENT

```
>> [x2, fs] = wavread('low.wav');
>> N = length(x2);
>> t = 0:1/fs:N/fs-1/fs;
>> subplot(2,1,1); plot(t, x2);

>> y = x2 - mean(x2);
>> y = y./max(abs(y));

>> subplot(2,1,2); plot(t, y, 'r');
```



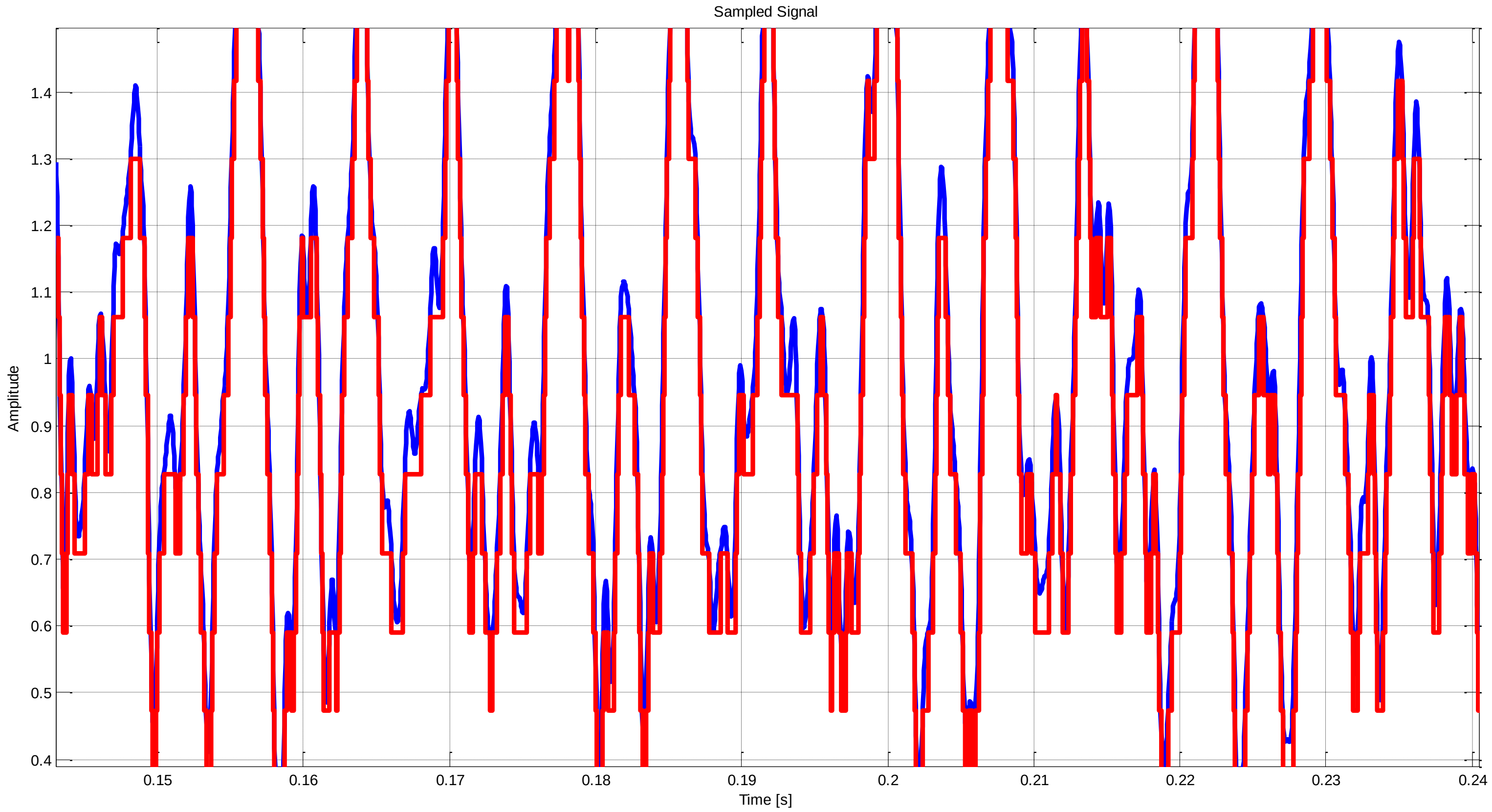


REDUCE BIT-RESOLUTION (4-BIT)

>> sampling3

>> sound(q_sig, fs)

>> sound(x2, fs)



SOUND RECORD (WIN)

dtype	Bits/sample
'double'	16
'single'	16
'int16'	16
'uint8'	8

Snytax

y = wavrecord(n, Fs, ch, ‘dtype’);

WAVRECORD Record sound using Windows audio input device.

Examples

Record 5 seconds of 16-bit audio sampled at 11025 Hz. Play back the recorded sound using wavplay. Speak into your audio device (or produce your audio signal) while the wavrecord command runs.

```
Fs = 11025;
y = wavrecord(5*Fs,Fs,'int16');
wavplay(y,Fs);
```

CONTENTS

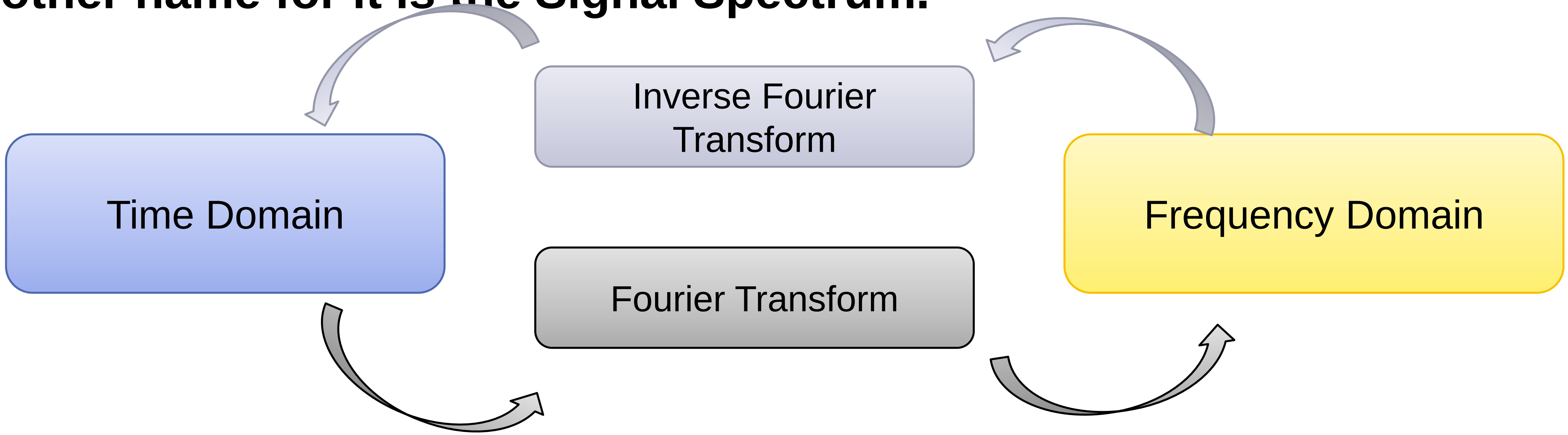
- ❑ Analog-to-Digital Conversion
- ❑ Sampling theory
- ❑ Time domain representation
- ❑ Fourier theory
- ❑ Frequency domain representation
- ❑ Feature extraction algorithms

CONCEPT OF FOURIER TRANSFORM (FT)

Since the concept summing up simple sine and cosine waves of any signal.

We look at the signal in Time domain.

The magnitude of signal from partial sum is called the Frequency Domain. Another name for it is the Signal Spectrum.



INTRODUCTION

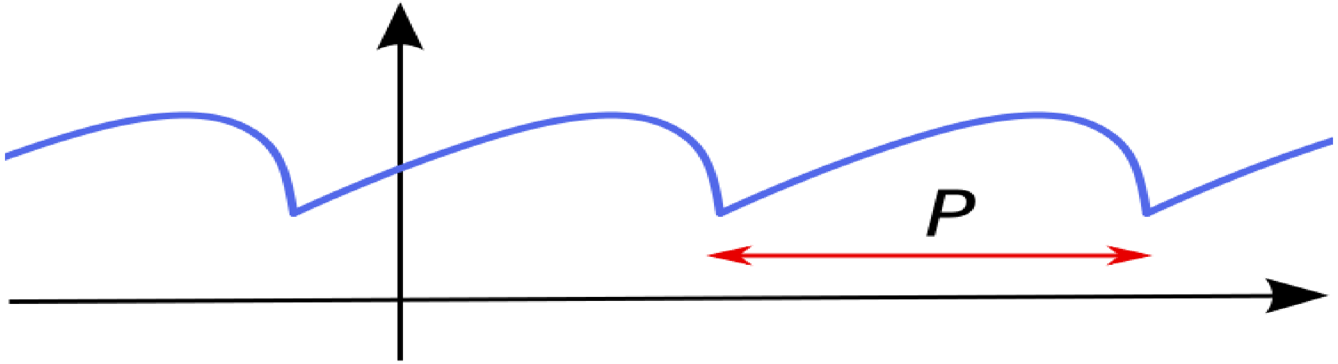
Who is Fourier?

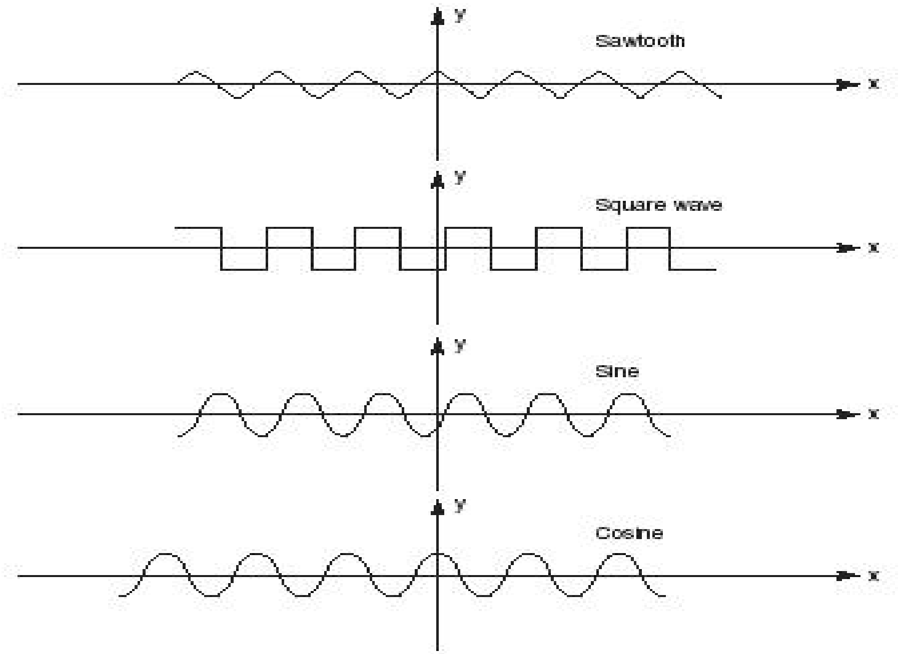
Jean Baptiste Joseph Fourier (March 21, 1768 - May 16, 1830).
He was a French mathematician and physicist.

Investigation of
Fourier Series, Heat Flow

INVESTIGATION OF FOURIER SERIES

What is a periodic function?

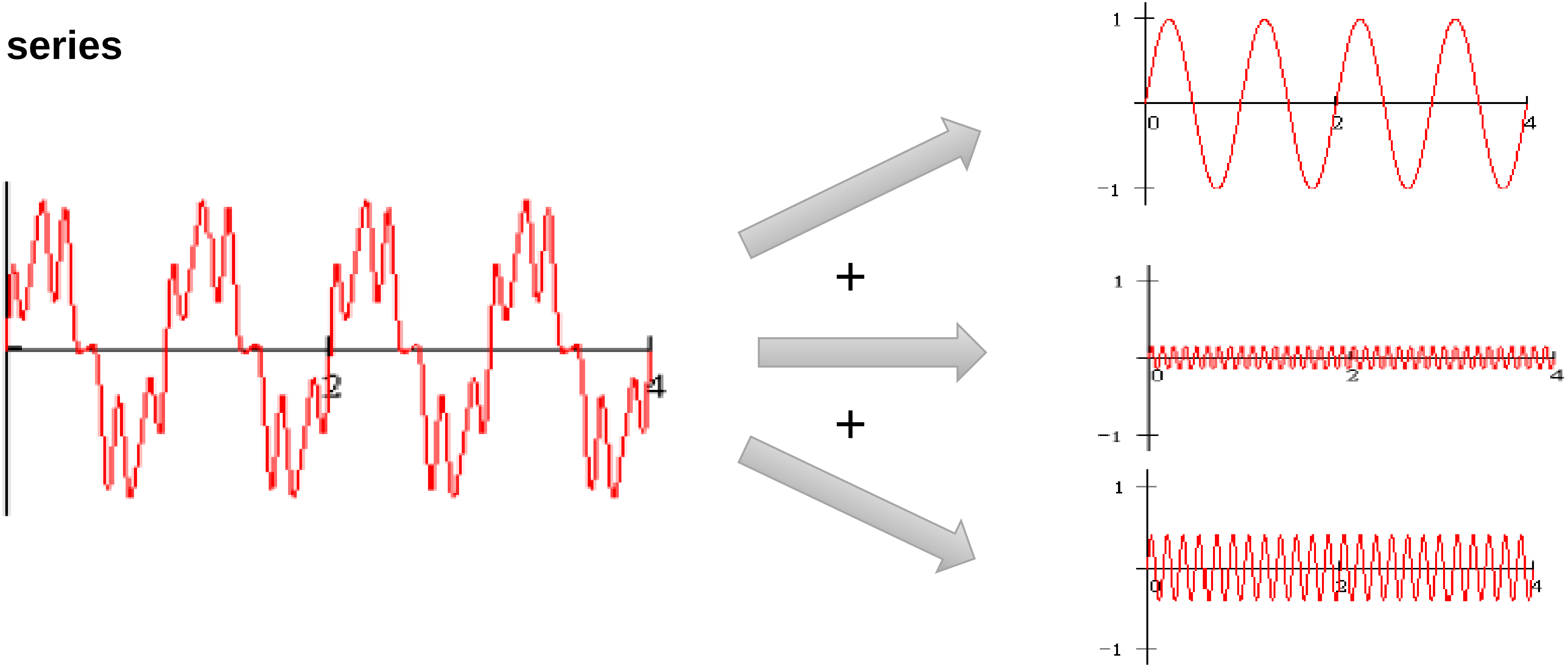




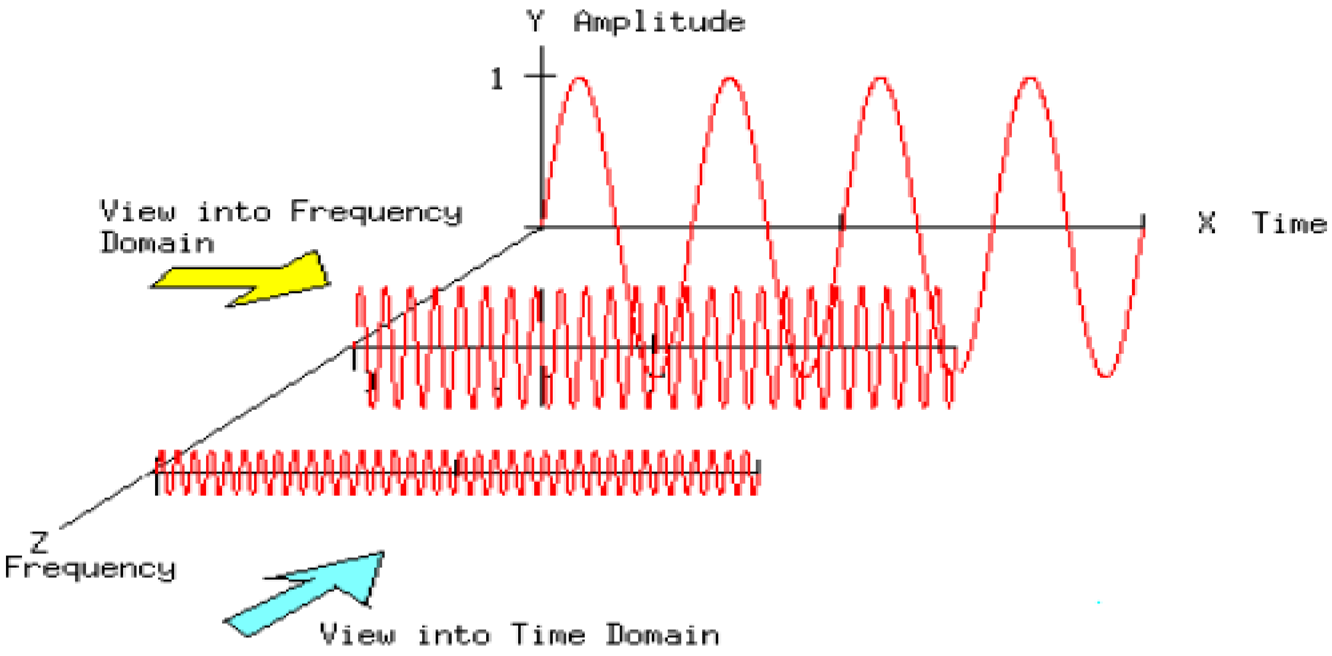
How can we represent it as the general periodic functions?

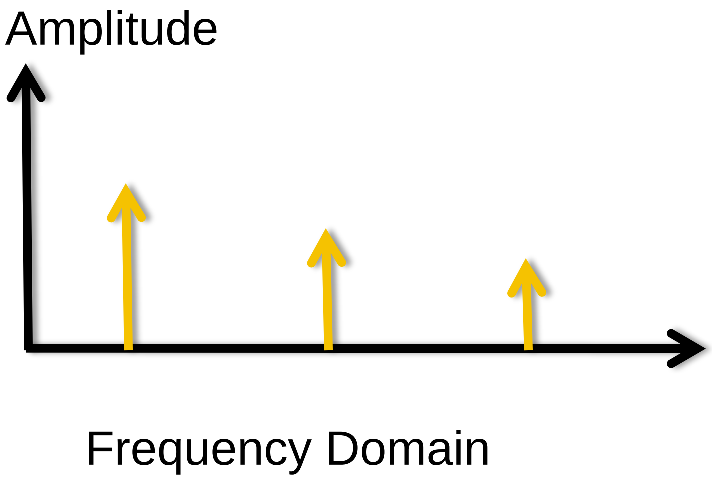
TIME DOMAIN

Fourier series



FREQUENCY DOMAIN





FOURIER TRANSFORM

Fourier transform can be defined by:

$$\mathcal{F}(f) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} \, dt$$

$$\mathcal{F}(f) = R(f) + iI(f)$$

MATLAB EXPERIMENT

```
>> myfft1
>> myfft2
```

CONTENTS

- Analog-to-Digital Conversion
- Sampling theory
- Time domain representation
- Fourier theory
- Frequency domain representation
- Feature extraction algorithms

APPLY FFT TO EXTRACT THE FREQUENCY COMPONENTS

- PDF of Piano sounds by Histogram Method
- Show the FFT analysis of Piano sounds
- Table of Peak Spectrum
- Classify the different Piano keys by FFT method

IMAGE PROCESSING BY MATLAB

REPRESENTING DIGITAL IMAGE

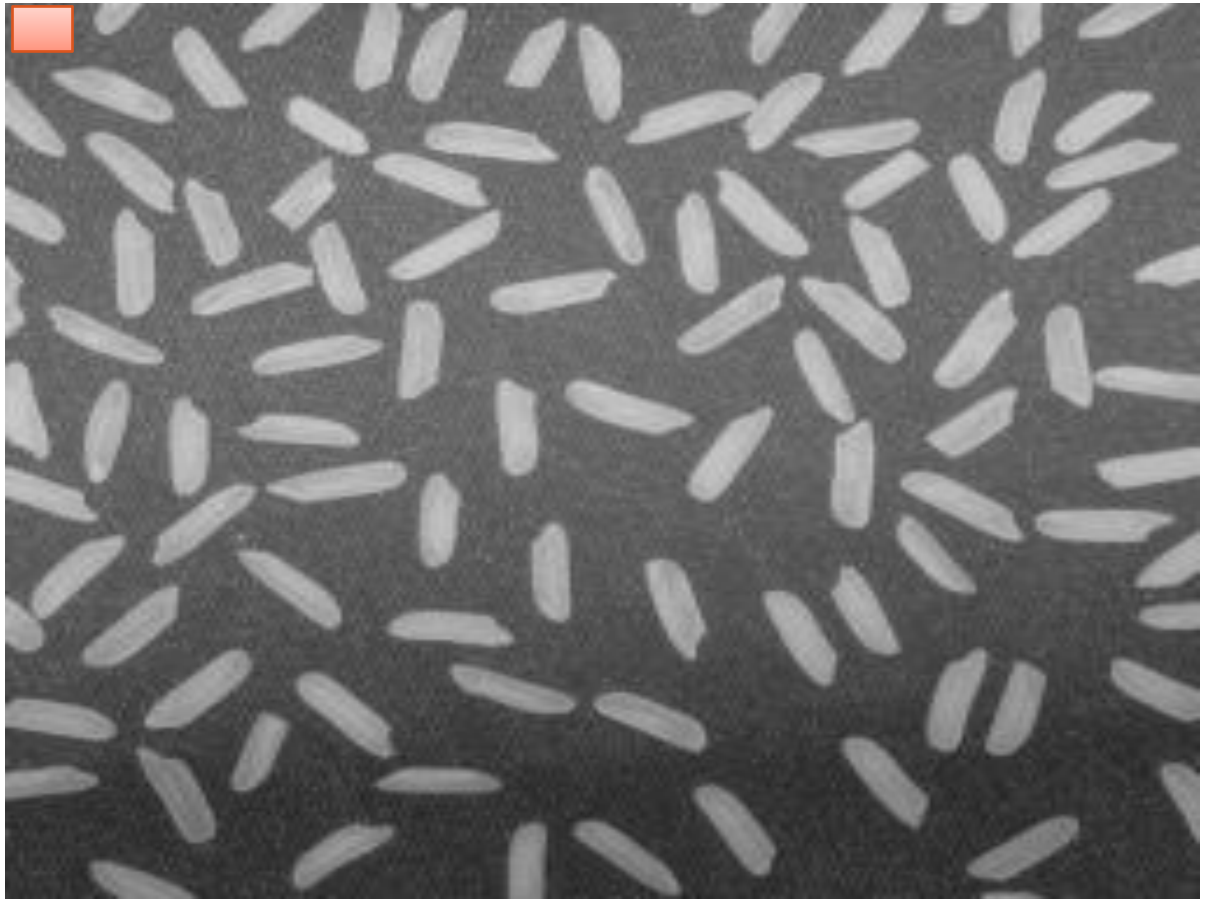
Digital image can be represented by 2-D array

f(x, y), M rows and N columns, coordinates x, y

Section of real plane is called "*Spatial Domain*"

2-D GRAY-SCALE IMAGE

Origin
f(0,0)



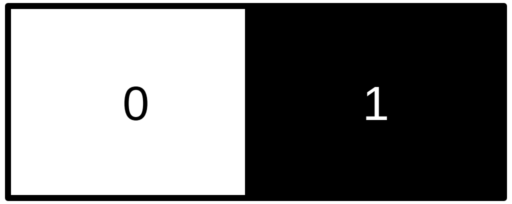
M

N

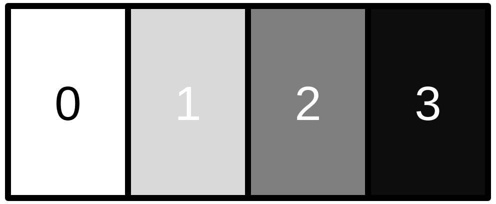
INTENSITY LEVEL

k-bit image $L = 2^k$

Quantized level in the interval [0, L - 1]



Binary scale



Gray scale

MATLAB COMMANDS

Basic commands

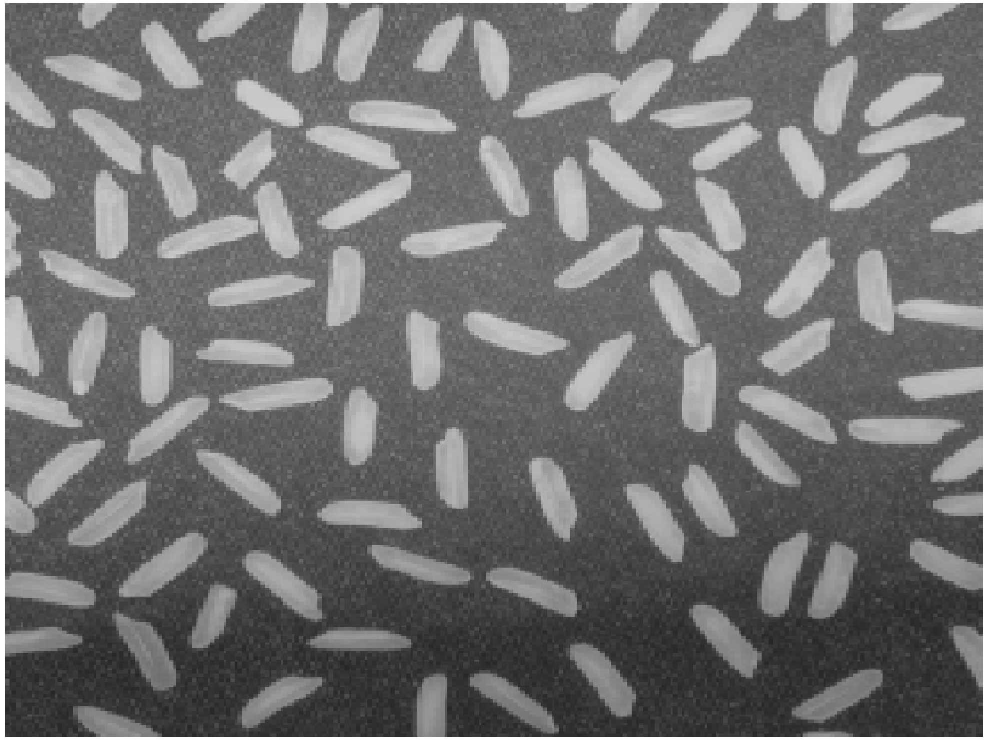
>> imread

>> imshow

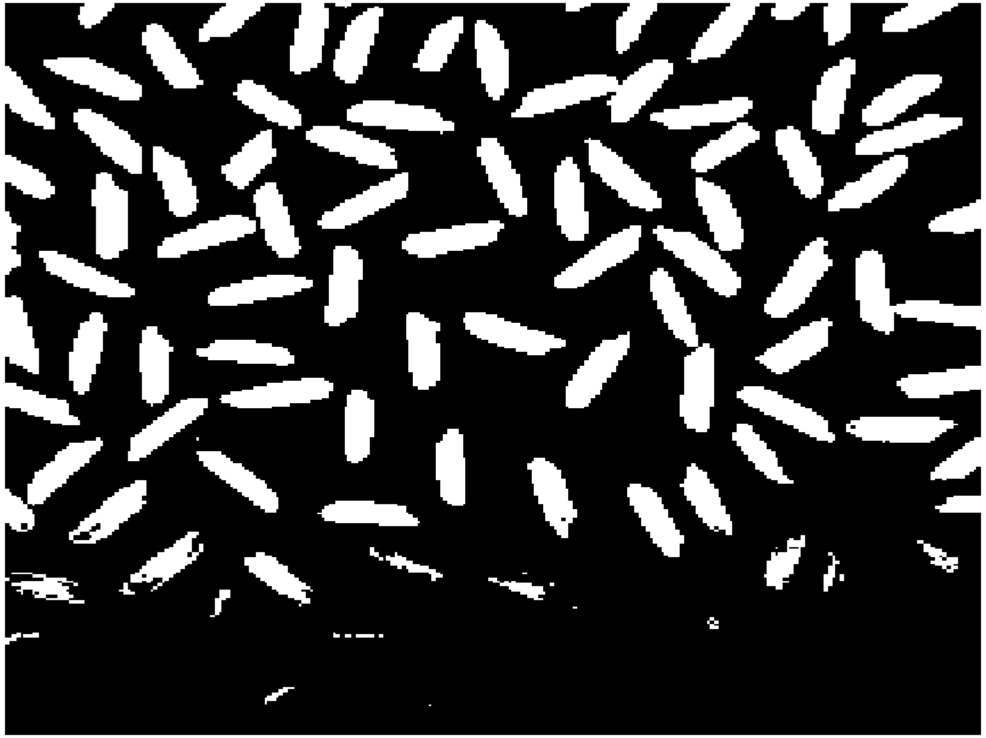
For example:

```
>> x = imread('rice.jpg');
>> imshow(x);
```

GRAY TO BINARY



Original image



Output image

ARITHMETIC OPERATIONS

Four basic operations are available (+, -, *, /)

Example, corrupted image by the addition of noise

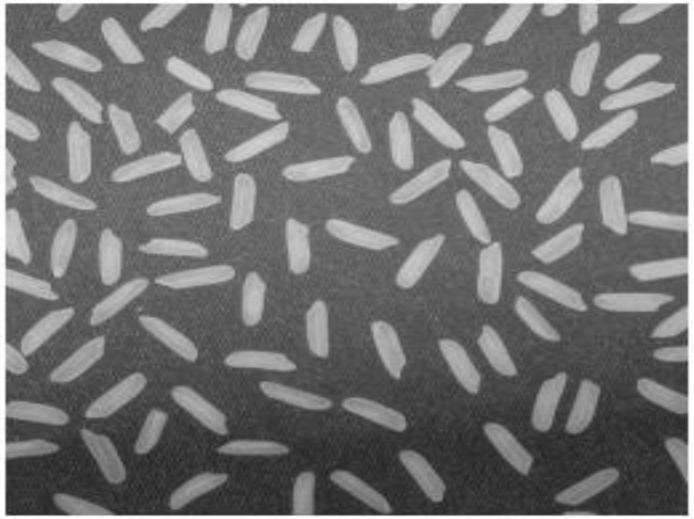
$$g(x,y) = f(x,y) + n(x,y)$$

where n is random noise function

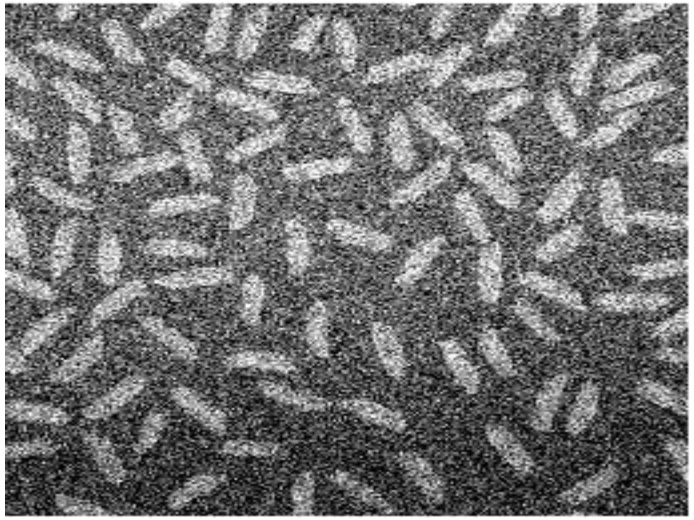
EXAMPLE OF ADDITIVE GAUSSIAN NOISE

MATLAB code

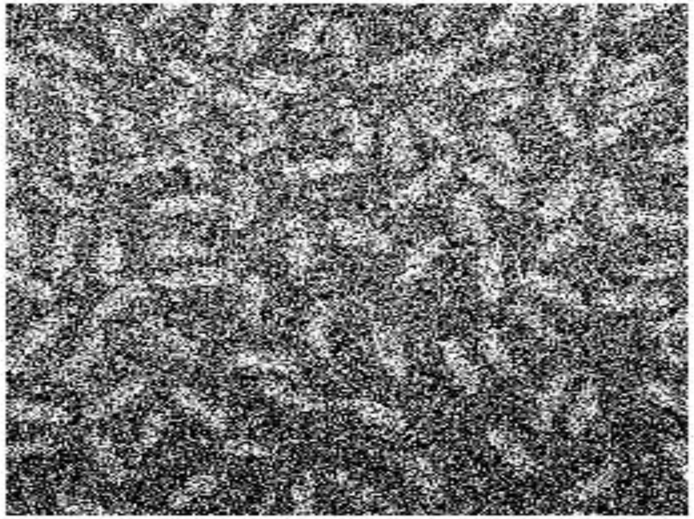
```
ns = randn(256,256);
nsimg = img + a*ns;
```



a = 1



a = 50



a = 100

SHADING CORRECTION

This is a normalization process of the image

$$f_m(x, y) = f(x, y) - \min\,\{f(x, y)\}$$

$$f_s(x, y) = \frac{(2^n - 1) f_m(x, y)}{\max\,\{f_m(x, y)\}}$$

MATLAB EXPERIMENTS

```
>> image1
>> image2
>> image3
```

COLOR IMAGE

RGB Color Model

CMY Model

HSI (Hue, Saturation, Intensity)

Color to Gray

RGB Luminance value = 0.3 R + 0.59 G + 0.11 B