BME 50500: Image and Signal Processing in Biomedicine

Lecture 1: Introduction, Digital Signals

Lucas C. Parra
Biomedical Engineering Department
City College of New York

CCNY

Content

Linear systems in discrete time/space
Impulse response, shift invariance
Convolution
Discrete Fourier Transform
Sampling Theorem
Power spectrum

Introduction to medial imaging modalities
MRI
Tomography, CT, PET
Ultrasound

Engineering tradeoffs
Sampling, aliasing
Time and frequency resolution
Wavelength and spatial resolution
Aperture and resolution

Filtering
Magnitude and phase response
Filtering
Correlation
Template Matching

Intensity manipulations
A/D conversion, linearity
Thresholding
Gamma correction
Histogram equalization

Python and A/D conversion (DAQ)
Recording signals

Stuff you need

Prerequisite:
BME 405, complex variables, Fourier series, some programming

Literature (selected book chapters)

Kayvan Najarian and Robert Splinter, Biomedical Signal and Image Processing, CRC Press, 2005.

Eugene N. Bruce, 2001. *Biomedical Signal Processing and Signal Modeling*. Wiley.

Semmlow, J.L., 2005. *Circuits, Signals, and Systems for Bioengineers: A MATLAB-based Introduction*. Elsevier Academic Press.

Jerry L. Prince & Jonathan Links, 2006. *Medical Imaging Signals and Systems*. Pearson.

Software

- Python or Matlab – we can select this together.

Notes and Slides

- These slides are not a complete record of the class material. They are only a brief reminder of what was discussed in class.
- Without careful notes you will not be able to study for your exams.
-

Grading

60% assignments, 15% midterm exam, 25% final exam

less than 60%: F
60% or more: D
70% or more: C
80% or more: B
90% or more: A
100% or more: A+

Exams and quizzes:

- Quizzes will test reading and programming homework (they are part of the 60%).
- Final exam will test everything.

Attendance:

- 100% attendance is expected (missing quiz = 0% credit)
- Homework assignments will include data recording which you complete independently using your own A/D board.
- If you miss class do not expect homework help from the TA.

Assignments

1. Python programming
    - Due one week from assignment.
    - Turn in by email **prior to class**!
    - May have pop quizzes on programming assignments.
2. Reading
    - Understand the subject and cover gaps.
    - May have pop quizzes on reading assignments.
4. Data recordings
    - Will schedule lab time during class hours
    - You will start during class time and complete recordings as assignments using your own A/D equipment at home.
    - Contact the TA if you have not yet received your own A/D board.
    - You must return this equipment at the end of the semester in order to receive a grade.

All written assignments must include: Name, date, course number, assignment number. Title whenever applicable. Form matters!

Getting ready - Computers

- All students must install python or matlab on their laptops.
- Instructions are here:
  https://www.parralab.org/teaching/signal-and-image/

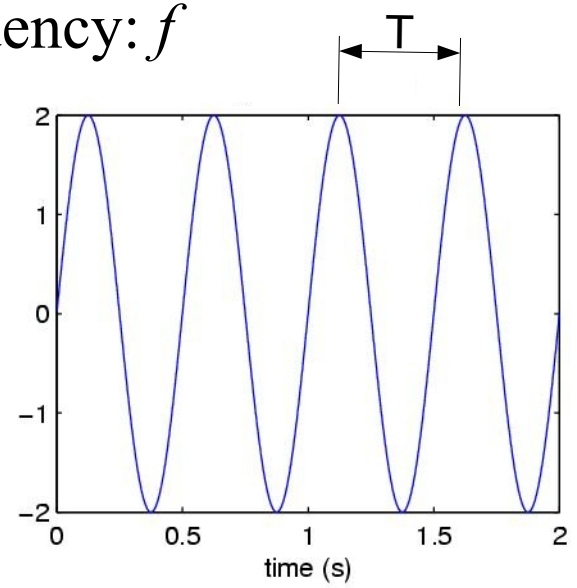
Digital signals

Digital signals are sampled and discretized

Cycle duration: T

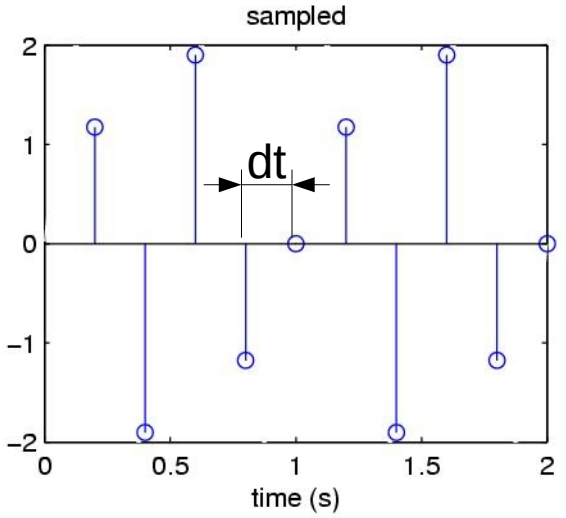
Frequency: f

$$f = \frac{1}{T}$$

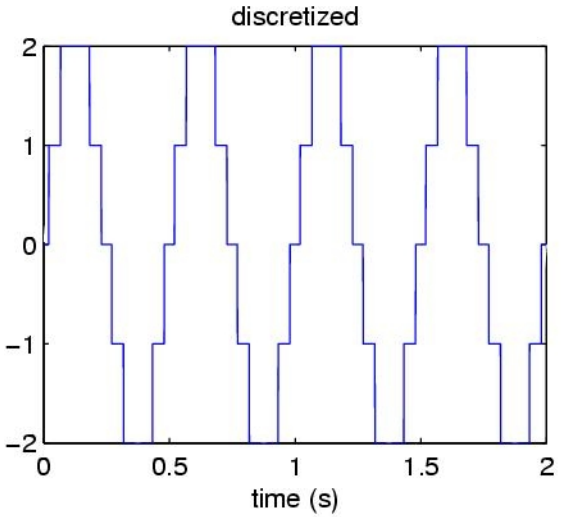


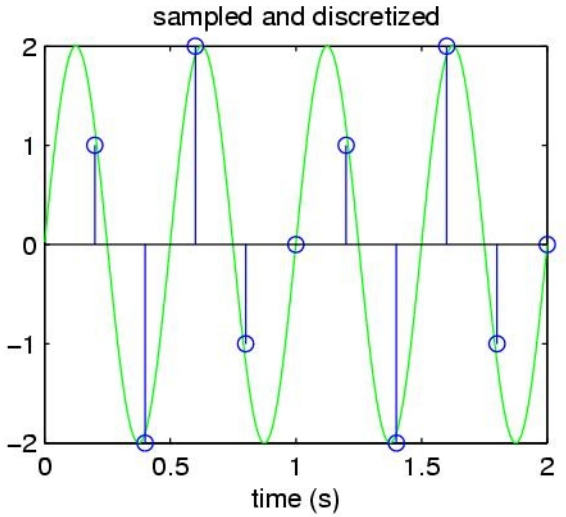
Sampling interval: dt

Sampling frequency: f_s

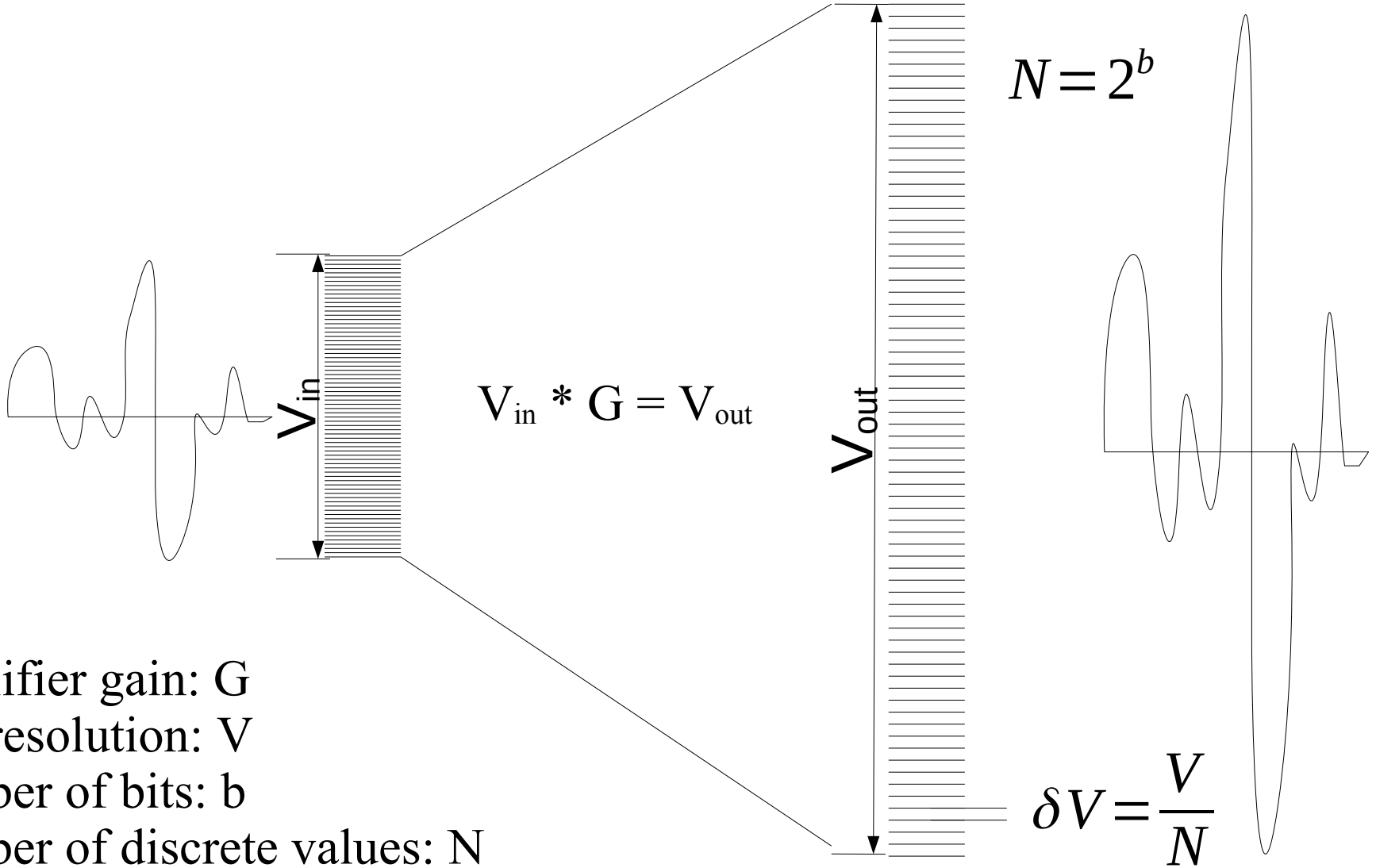


$$f_s = \frac{1}{dt}$$





Discretization with amplification



Amplifier gain: G
A/D resolution: V
Number of bits: b
Number of discrete values: N
Signal dynamic range: V_{in}
A/D dynamic range: V_{out}

Typical sampling and discretization values

$$N = 2^b$$

Audio:
b = 16 per sample (16 bit)
f_s = 44100Hz
Two channels: stereo
Data type: int16 (-32767 … 0 … +32768)

Images:
b = 8 per pixel, per color
3 colors RGB
Data type: uint8 (0 … 255)

Video:
f_s = 30 Hz (frames per second)

3 bit binary	1 digit decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Discretization

2 gray levels

4 gray levels

8 gray levels

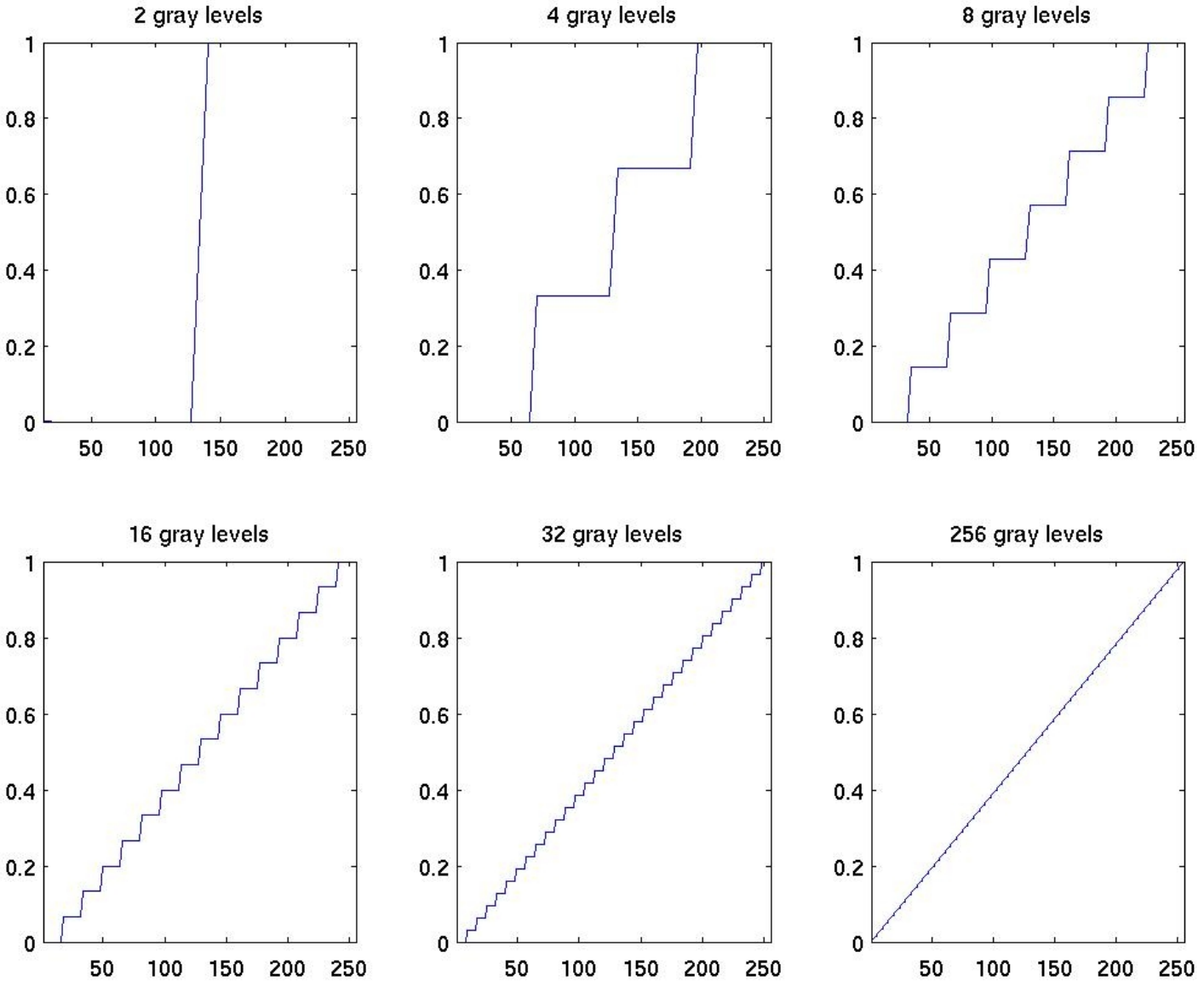
16 gray levels

32 gray levels

256 gray levels

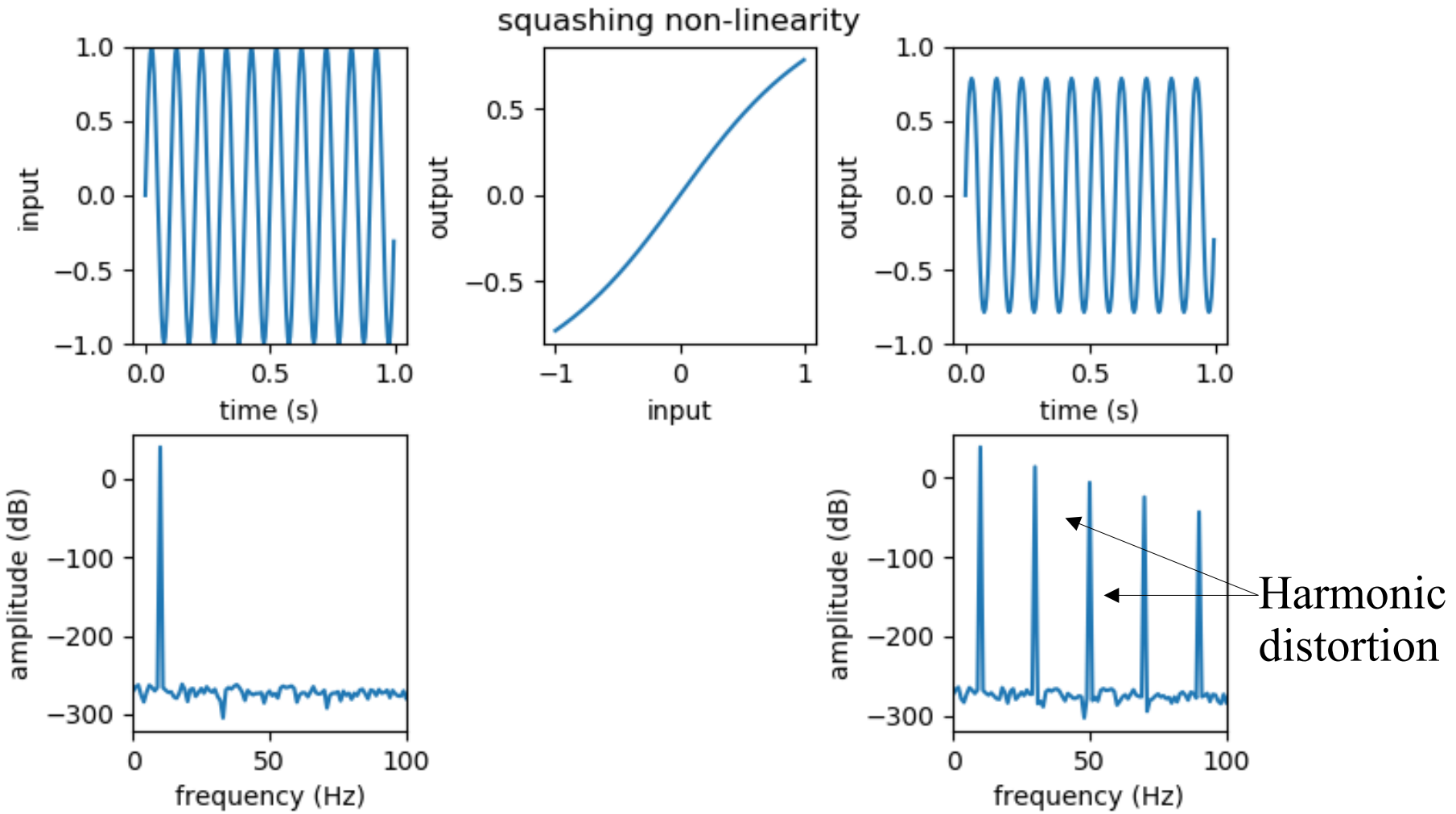
Discretization

The problem with discretization is that it is a non-linear process.



Non-linear Amplification

Non-linearity creates artifacts in the *frequency domain*.



Discretization

Harmonic distortion explained ...

For example distortion of quadratic non-linearity leads to frequency doubling:

$$x(t)=\sin(\omega t)$$

$$y(t)=x^2(t)=\sin^2(\omega t)=\frac{1}{2}-\frac{1}{2}\cos(2\omega t)$$

Cubic leads triple frequencies:

$$y(t)=\sin^3(\omega t)=\frac{3}{4}\sin(\omega t)-\frac{1}{4}\sin(3\omega t)$$

General non-linearity contains all orders according to Taylor expansion:

$$y=f(x)=\sum_{n=0}^{\infty}\frac{1}{n!}\left[\frac{\partial^n f(x)}{\partial x^n}\right]_{x=0}x^n$$

Sampling

If the signal is sampled every Δt (in seconds) the sampling frequency (in Hz=s^{-1}) is given by:

$$f_s = \frac{1}{\Delta t}$$

For images the resolution is given by pixel size Δx (in cm) or as pixels per cm:

$$f_s = \frac{1}{\Delta x}$$

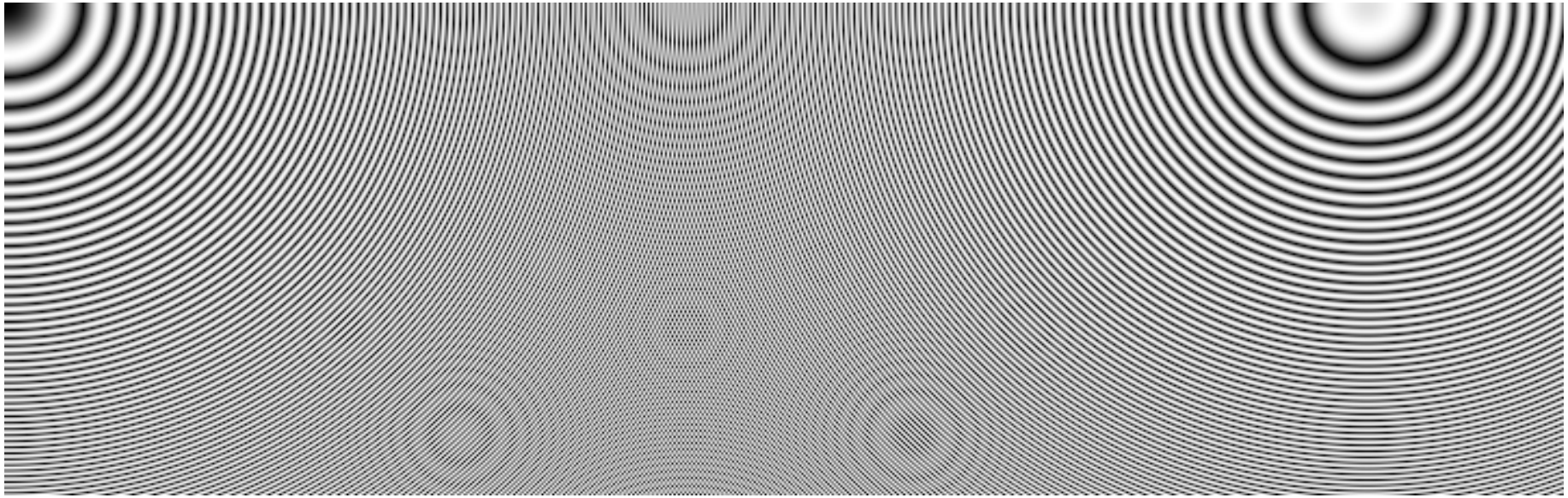
The sampling theorem states that one has to sample a signal at least twice the frequency of the highest oscillation.

Therefore a 10Hz oscillation has to be sampled at least at every 1/20s.

Sampling

If we violate that requirement we get aliasing:

Example: $f(x,y) = \sin(x^2 + y^2)$. 720x240 pixels and amplitude scaled to [0 255] and discretized.



Sampling creates artifacts in the *space (time) domain*.

Assignment 1: Generate figures in slide 7, 8 and 13 .

Some python commands used in class

```
import numpy as np
import matplotlib.pyplot as plt

pi = np.pi

t = np.arange(2*fs)/fs # time vector

plt.clf()
plt.subplot(2,2,1)
plt.plot(t,y)
plt.stem(t,y)

P=plt.imread('penny.jpeg').copy()
P[:,:,[0,2]]=0
plt.imshow(P)
```

```
P=np.mean(P,2)
for x in np.arange(0,10):
    y=x
    plt.clf()
    crop = P[y:y+40,x:x+40]
    ax=plt.imshow(crop,cmap='gray')
    plt.pause(1/30)
    plt.show()

x=np.tile(np.linspace(0,1,700),(400,1))
y=x.T
r2=x**2+y**2
```

Programming Assignments

- Assignments are due within one week and are submitted per email to bme505@gmail.com.
- **If you do not submit on time, you will get 0 (zero) credit.**
- If there is a medical reason why you need extra time, you must ask for an extension before the deadline.
- **Requests to extra time after the deadline will be declined.**
- You are **not** allowed to collaborate on programming assignments. If you do, I twill look like copying which is strictly forbidden.
- **If you copy code, both the author and the copier will get zero credit.** Therefore, do not email each other programs, do not use USB drives to copy code, do not screen share code.
- Be sure to write comments in your code to explain what you are doing.
- Submit single executable file called: first_last_number.py
- Do not submit image files or text files. Only submit data if that is part of the assignment.
- Your program should not generate warnings.
- Programs that crash with an error message will receive 0 (zero) credit.
- Your program must load all required data. Assume that data files are in the current directory. All required data will be posted on the web, unless otherwise stated.
- The criteria for full credit should be clear. If not, please ask in class. Do not take chances by assuming that your work is "sort of correct".