Course Logistics
Review Core DSP Concepts
Audio Feature Extraction

Course Logistics

Goals of Course

- **Gain an appreciation for the applications of signal processing**
- **Learn to implement signal processing algorithms in Python**
  - Develop good coding habits
- **Learn a bit about audio and image processing, two important areas where signal processing is critical**
- **Introduce: machine learning and optimization**

Course Logistics

- **Basics**
  - Lecture: Monday 4:00pm to 5:15pm in 1B32
  - Lab clinic one: Monday after lecture in 1B32
  - Lab clinic two: Wednesday 3:00pm in ECEE 281/282
  - Office hours: by appointment, before/after lecture/clinic
  - My email: michael.perkins@colorado.edu
- **Canvas**
  - Labs and lectures will be posted here
  - Assignments will be submitted here
  - Syllabus and lab template are in the “course documents” module

Course Organization

- **The course consists of 6 labs**
  - Labs are either 2 or 3 weeks in duration
  - You will write a report and submit a pdf on Canvas
    - ✓ Follow the lab report template format
  - No final
- **Lab topics**
  - Music feature extraction (2 labs)
  - Image processing (1 lab)
  - MP3 and JPEG compression (2 labs)
  - Machine Learning / Optimization (1 lab)

Grading

- **From the syllabus**

Grading

Your deliverable for each lab will be a report. A grading rubric will be distributed for each lab explaining how points are allocated for that report. The quality of your code is an important part of your grade. Your final grade will be based on the following as manifested by your reports and code:

A – Exceptional mastery of the course material and concepts. Clear, well organized, and essentially error-free reports (both technically and grammatically) that go the extra mile. "A" work is more than just complete and adequate, it is impressive.

B – Adequate understanding of nearly all the concepts covered in the course. Well organized and proficiently written reports. "B" work is both complete and respectable.

C – Reasonable effort was put into the class, however, substantial gaps in conceptual understanding exist and/or the reports are poorly organized, perfunctory[1], or contain significant errors (either technical or grammatical).

D – Little effort was put into the class, and little was learned.

F – Minimal attendance or attempt was made at mastering the material and completing the labs.

Python

- **The labs in this course will be implemented in Python**
- **Why Python?**
  - Free
  - Widely used in industry for applications other than scientific computing
    - ✓ Many of you will encounter it whether you do DSP or not in your job
  - Supported by a rich set of libraries
  - Has Matlab-type functionality
  - The de-facto language for machine learning
  - Continues to grow in popularity

Python Installation

- **Make sure to install Python 3.x do NOT use a flavor of python 2 (it is obsolete).**
- **Installation website:**
  - https://www.python.org/downloads/
- **IDE suggestions can be found at**
  - https://realpython.com/python-ides-code-editors-guide/
- **You will also need to install four packages (e.g using pip/pip3). These will be used extensively. They provide matlab type functionality**
  - Numpy
  - Scipy
  - Matplotlib
  - Pillow

Getting Started with Python

- **Download the *python tutorial* pdf from Canvas**
  - It is in the course documents module
  - It contains code samples that cover much of what you will need for this class
  - It contains links to other references
- **Huge amounts of information and examples are available by googling**
- **Don't delay! You should have python and the required modules installed by Wednesday's lab clinic**

Python Awareness

- **Python basics**
  - Python is interpreted, not compiled
  - Variable types are automatically determined, not declared
  - Standard syntax structures are supported
  - Indentation matters. It is used in place of brackets { } to determine scope
  - Array indexing starts at 0
  - Matlab type functionality is available through libraries
    - ✓ Numpy, Scipy, Matplotlib

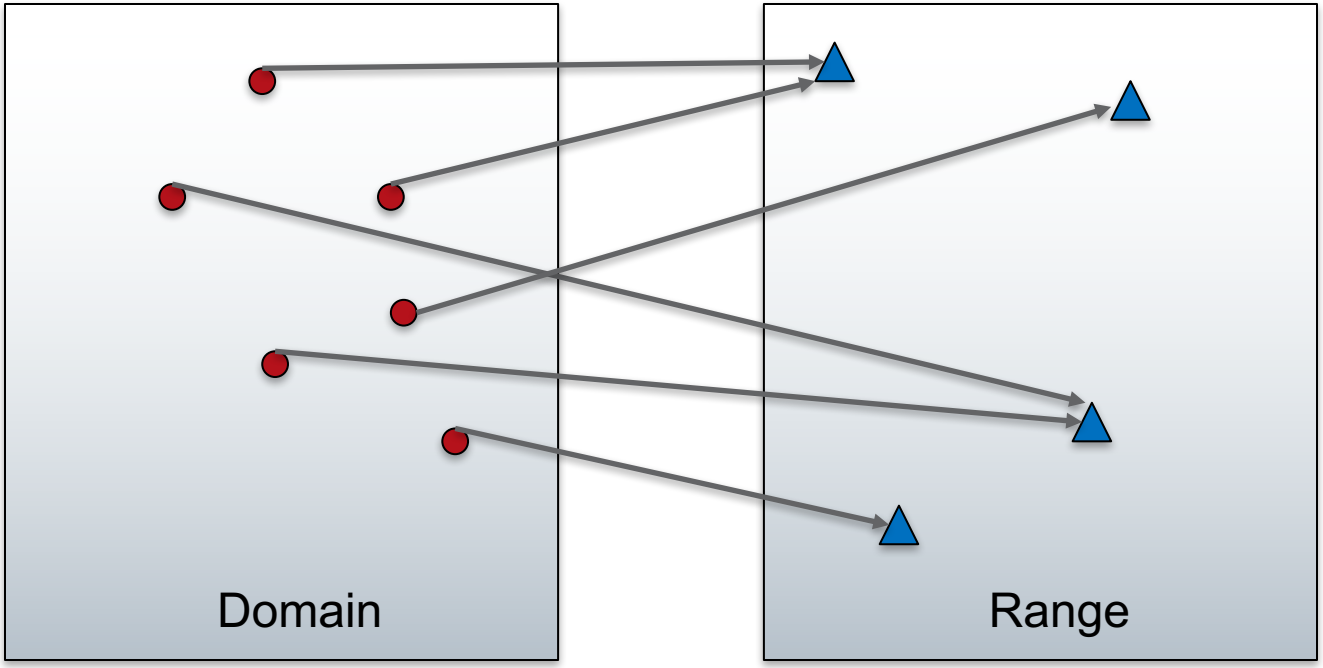
Numpy

- **Facilitates vector operations**
- **Try to do things in vector format when possible**
  - Often possible to write code compactly and avoid for( ) loops
    - ✓ Speeds up execution
  - Component-by-component operations are supported
- **Good numpy functions to learn immediately:**
  - numpy.arange(start, stop, step)
  - numpy.linspace(start, stop, num_points)
  - numpy.zeros(vector_size, numpy.int16)
    - ✓ Other types, such as float64, int32, uint8, etc. are supported
  - numpy.asarray(python_list, numpy.float64)
    - ✓ Other types, such as float64, int32, uint8, etc. are supported

Review Core DSP Concepts

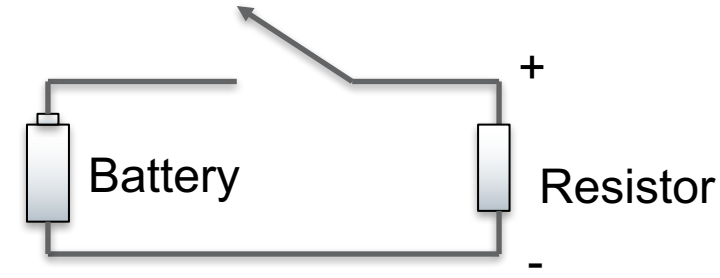
What is a Function?

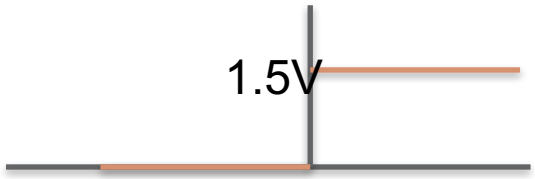
- **Function**
  - A **function** is a mapping from one set (the domain) to another set (the range). Each member of the domain is mapped to only one member of the range



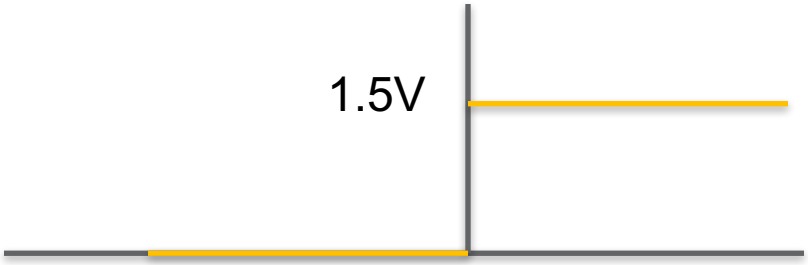
What is a Signal?

- **Signal**
  - A **signal** is a function that describes the behavior of a natural or man-made phenomena
- **Example**
  - The voltage appearing across a resistor when a switch is closed

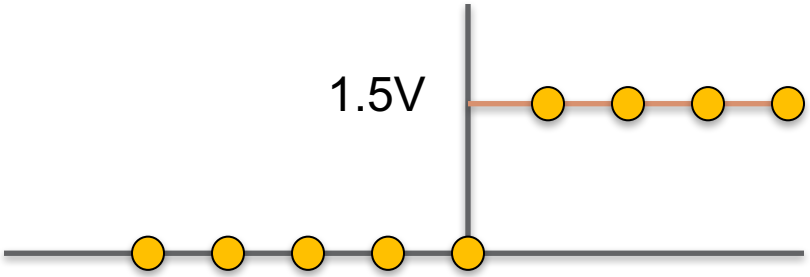




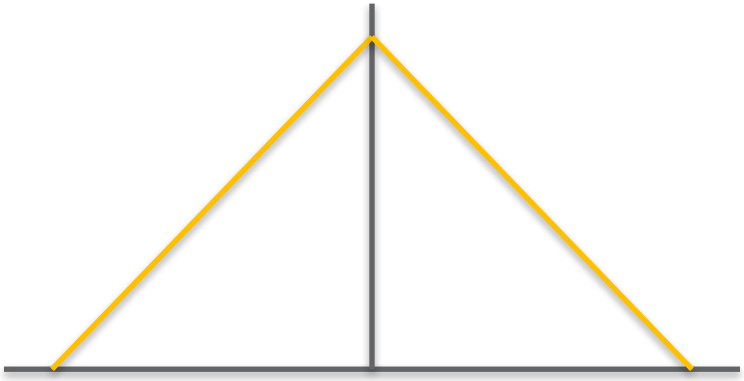
What is Discrete Signal?



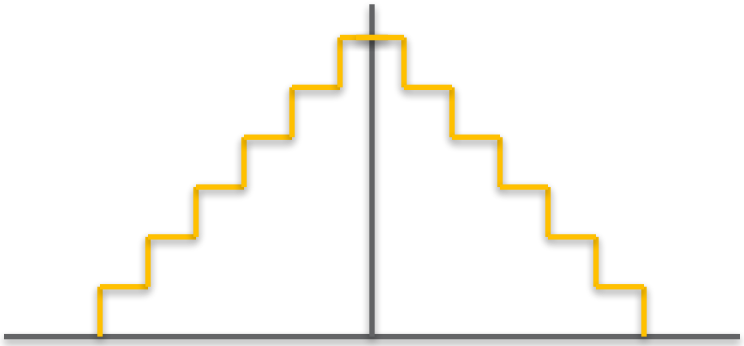
Continuous time



Discrete time



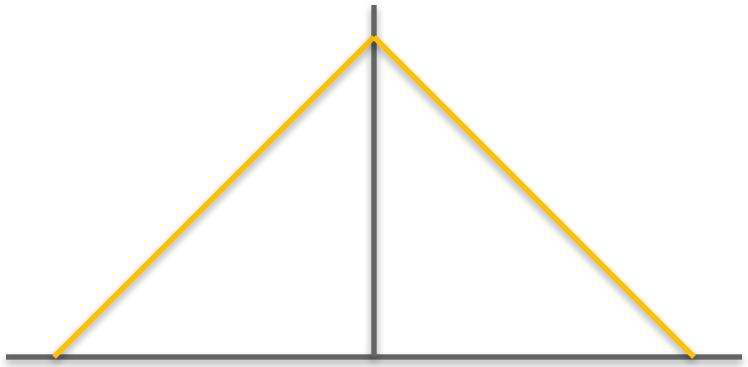
Continuous value



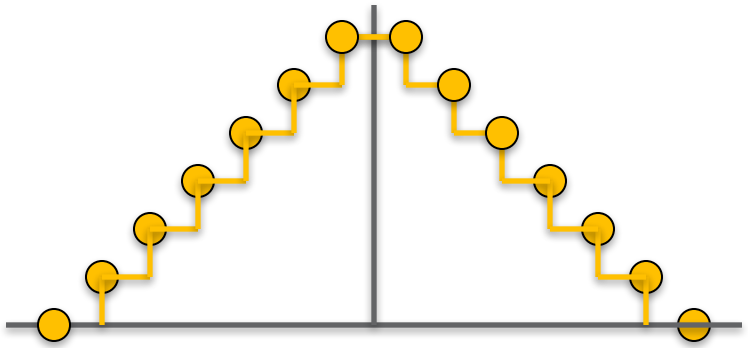
Discrete value

What is a Digital Signal?

- **We will define a "digital signal" to be one that is both discrete time and discrete value**
  - The only type of signal we can process on a computer!
  - Note that "digitization" distorts a continuous signal (e.g. the peak is missing below)



Continuous time and continous value



Digital signal (discrete time and value)

Scalars and Vectors

- **Scalar: a single number**
  - examples: $1, \pi, 2.8765, \dfrac{346}{789}$
- **Vector: a 1-D array of scalars**
  - example:

$$\underline{\mathbf{x}} = \begin{bmatrix} 7.2 \\ e \\ -24/9 \\ 0 \\ \sqrt{2} \end{bmatrix}$$

Column and Row Vectors

- **Vectors are traditionally visualized as columns**
- **If we mean a row vector, we use the "Transpose" notation below**

$$\underline{\mathbf{x}}^T = \begin{bmatrix} 7.2 & e & -24/9 & 0 & \sqrt{2} \end{bmatrix}$$

- **We put an underscore underneath a vector (and/or make it bold) to reduce confusion**
  - Scalars will be lower case, not bold, and not underlined

Matrix

- **A matrix is a 2-D array of numbers**
  - example:

$$A = \begin{bmatrix} 1 & 8 & 9.2 & -24/5 \\ 2.3 & e & \pi & -\sqrt{5} \\ 14 & 2.4 & -0.98 & 16 \\ 3^{2.2} & 5.4 & -2.3 & .99 \end{bmatrix}$$

  - We will use capital letters to denote matrices

Matrix as a vector of vectors

- **Can write the previous matrix as follows**

$$X = \begin{bmatrix} \underline{\mathbf{x}}_1 & \underline{\mathbf{x}}_2 & \underline{\mathbf{x}}_3 & \underline{\mathbf{x}}_4 \end{bmatrix}$$

Where in our example,

$$\underline{\mathbf{x}}_1 = \begin{bmatrix} 1 \\ 2.3 \\ 14 \\ 3^{2.2} \end{bmatrix} \quad \cdots \quad \underline{\mathbf{x}}_4 = \begin{bmatrix} -24/5 \\ -\sqrt{5} \\ 16 \\ .99 \end{bmatrix}$$

Matrix transpose

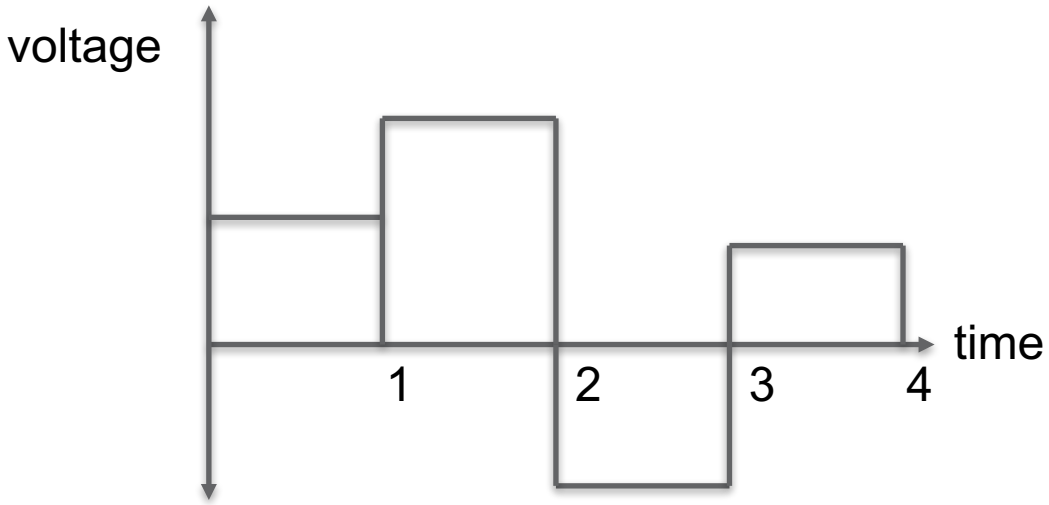
- **The transpose of this matrix is**

$$X^T = \begin{bmatrix} \underline{\mathbf{x}}_1^T \\ \underline{\mathbf{x}}_2^T \\ \underline{\mathbf{x}}_3^T \\ \underline{\mathbf{x}}_4^T \end{bmatrix} = \begin{bmatrix} 1 & 2.3 & 14 & 3^2.2 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ -24/5 & -\sqrt{5} & 16 & .99 \end{bmatrix}$$

- **The first column becomes the first row, the second column the second row, etc.**

Energy of a Digital Signal (Vector)

- **We define the "energy" of a digital signal by imagining each vector component is a voltage applied to a one-ohm resistor for 1 second**



Energy of a Digital Signal cont.

- **The power of a constant voltage, *v*, applied to a resistor, *R*, is $P=v^2/R$. Energy is "power times time", so for $R = 1$, over a one second interval, each sample delivers the energy $x^2(n)$. The total signal energy is therefore:**

$$\mathcal{E}(\underline{\mathbf{x}}) = \sum_{n=1}^{N} x^2(n)$$

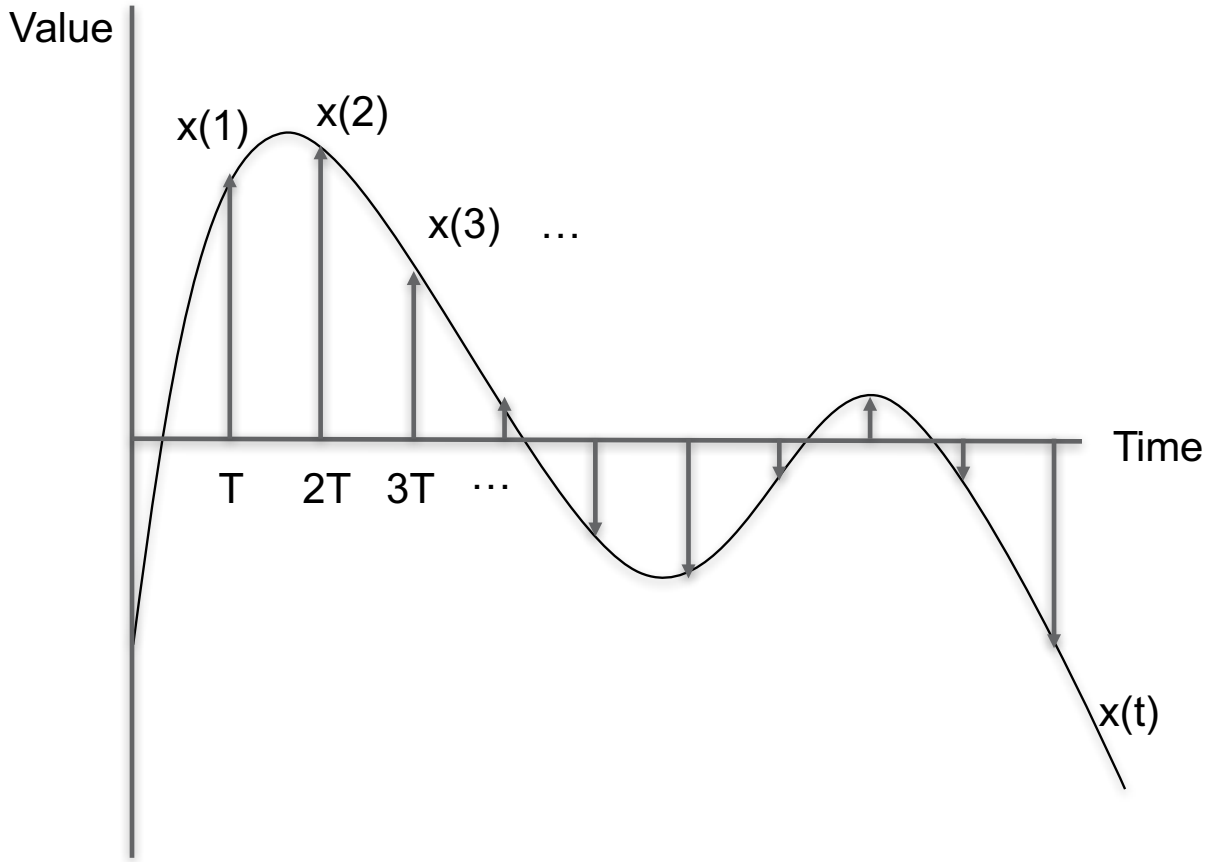
Power of a Digital Signal (vector)

- **Power is the energy delivered per unit time, so we get the average power of a vector by dividing its energy by *N***

$$\mathcal{P}(\underline{\mathbf{x}}) = \frac{1}{N}\mathcal{E}(\underline{\mathbf{x}}) = \frac{1}{N}\sum_{n=1}^{N} x^2(n)$$

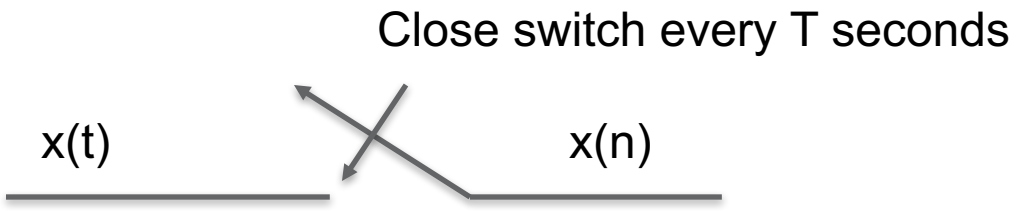
Sampling continuous time function

- **Assume we instantaneously measure a signal's value every T seconds**



Sampling cont.

- **Sampling is converting a continuous time signal to a discrete time signal (vector)**
  - We intentionally suppress the dependency on the sampling period, T, when labeling samples



- **Sampling Frequency**

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

Sampling and Vectors

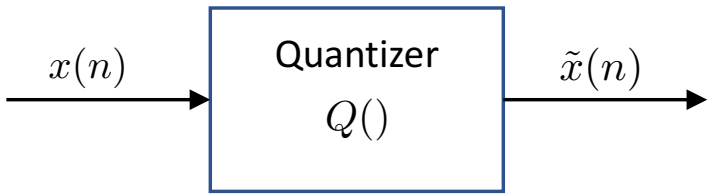
- **Each component of the signal vector is one sample**
  - It is customary to treat each sample as "one second" apart from its neighbors, even though the real interval is $T = 1/F_s$
  - A scaling is applied at the end to "fix things up"
- **Long vectors are frequently sub-divided into a series of sub-vectors (aka "frames" or "blocks")**
- **Many interesting signal statistics are computed on a frame-by-frame basis (e.g. mean, variance, etc.)**
  - Note that a digital image, when row-scanned, is a vector. Video is a long vector, video frames are sub vectors

Sampling cont.

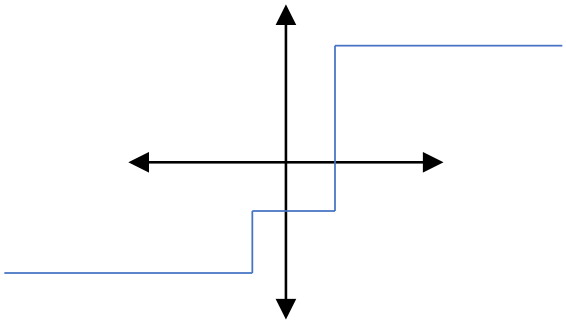
- **Signals that change quickly must be sampled faster (use a higher *Fs*)**
- **If we don't sample fast enough, we experience a distortion called aliasing**
  - We'll look at this in the homework
- **The Nyquist theorem tells us how fast we have to sample**

Quantization

- **Sampled values must be "quantized" to a finite number of levels**
- **Quantizing is just rounding off**

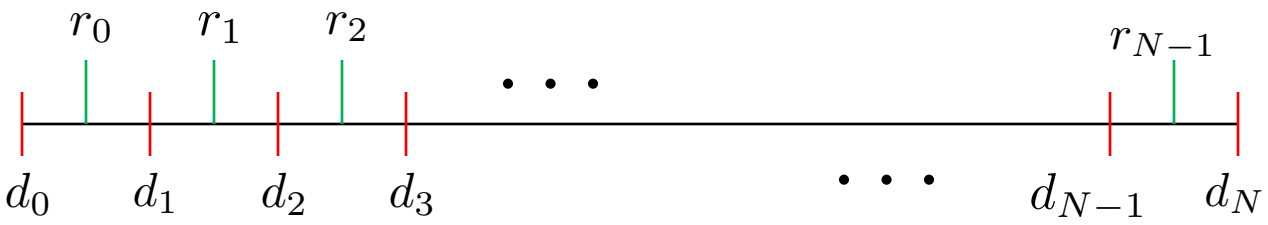


3 level non-uniform quantizer



Uniform Quantizer

- **Values between two decision levels are rounded off to a reconstruction level between them**
- **Decision levels and reconstruction levels are uniformly spaced**



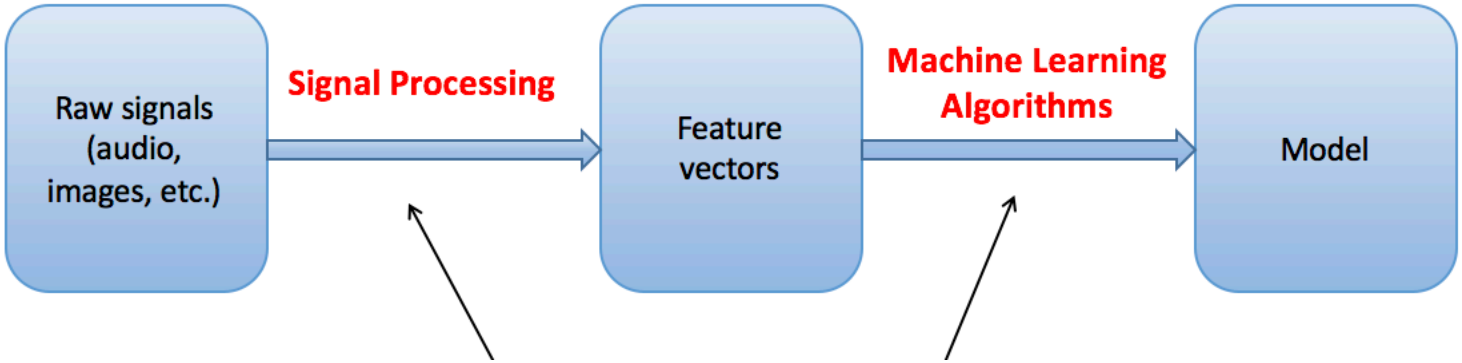
Lab 1
Audio Feature Extraction

Lab 1 Motivation

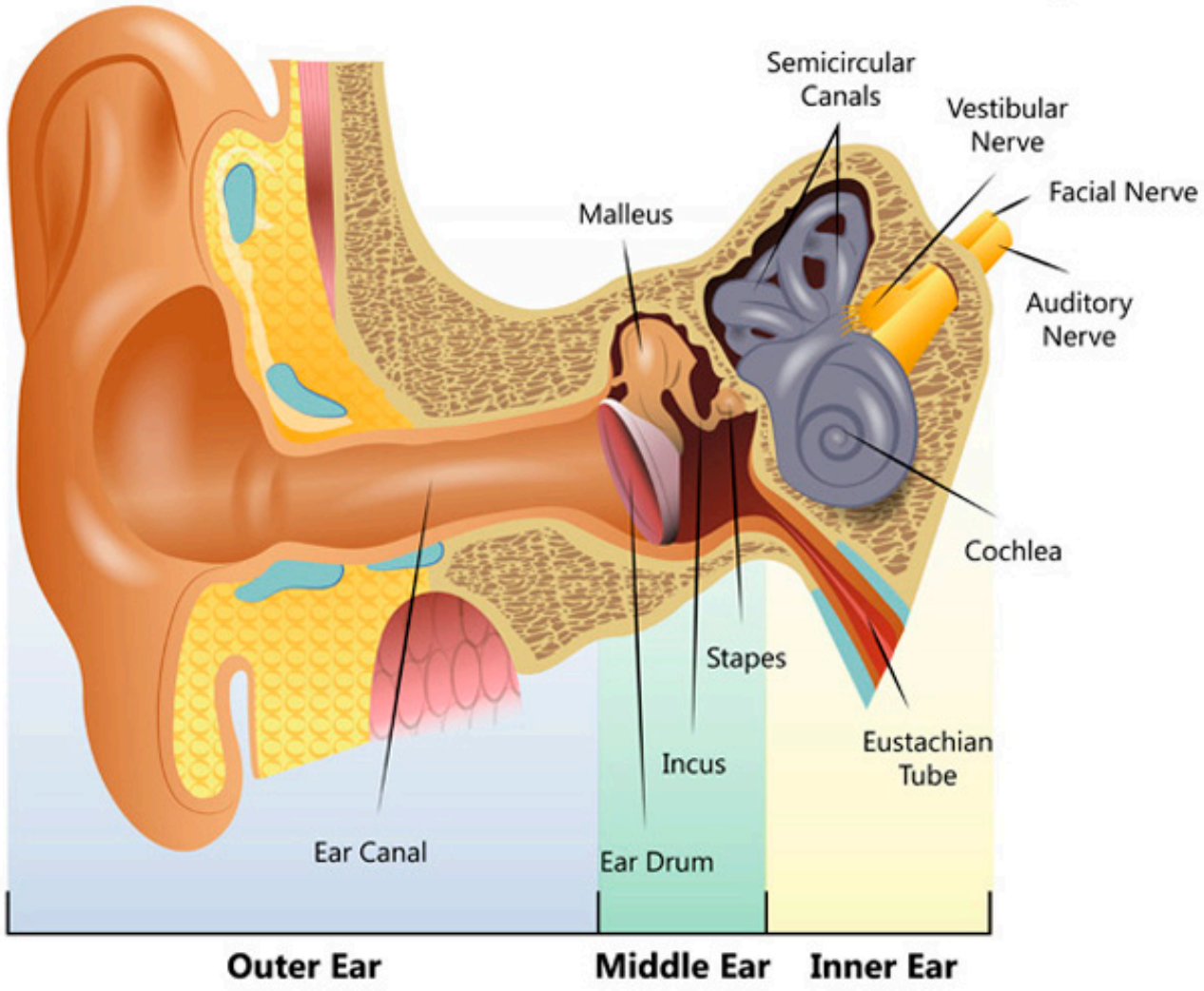
- **We want to analyze music, e.g., to determine in which genre it lies**
  - Blues, rock, jazz, classical, etc.
- **Strategy**
  - Compute a vector of “features” (aka “descriptors”)
  - See if the feature vectors form clusters that correspond to genre
    - ✓ An example of machine learning
- **Goals for lab 1**
  - Compute low-level features of interest
  - Begin developing python proficiency

The big picture

- **High-level diagram of a classification system**
  - Extract the features needed for music genre classification
  - Analyze the feature vectors and assign a class



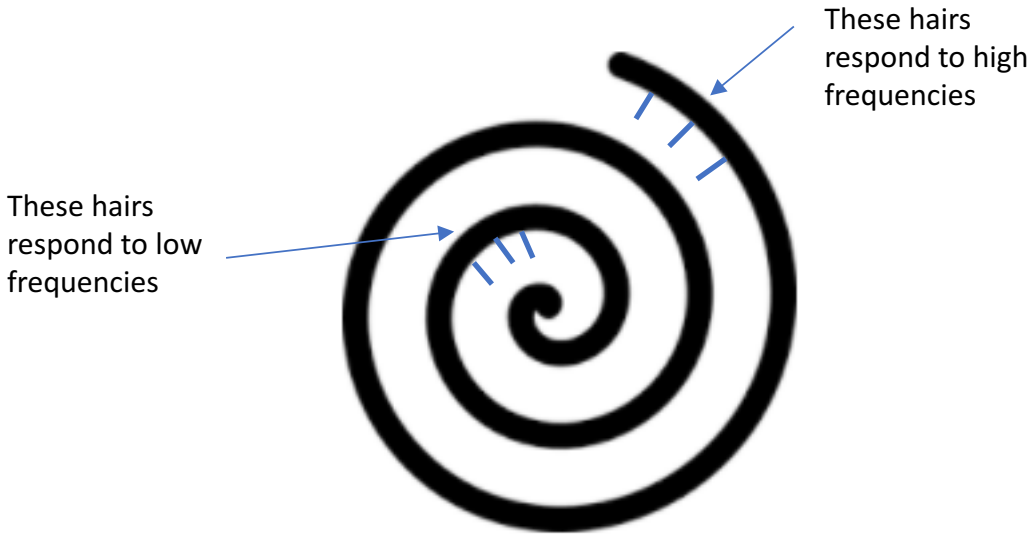
Human Ear


Semicircular Canals
Vestibular Nerve
Facial Nerve
Malleus
Auditory Nerve
Cochlea
Stapes
Eustachian Tube
Incus
Ear Canal
Ear Drum
Outer Ear
Middle Ear
Inner Ear

The Ear's Processing

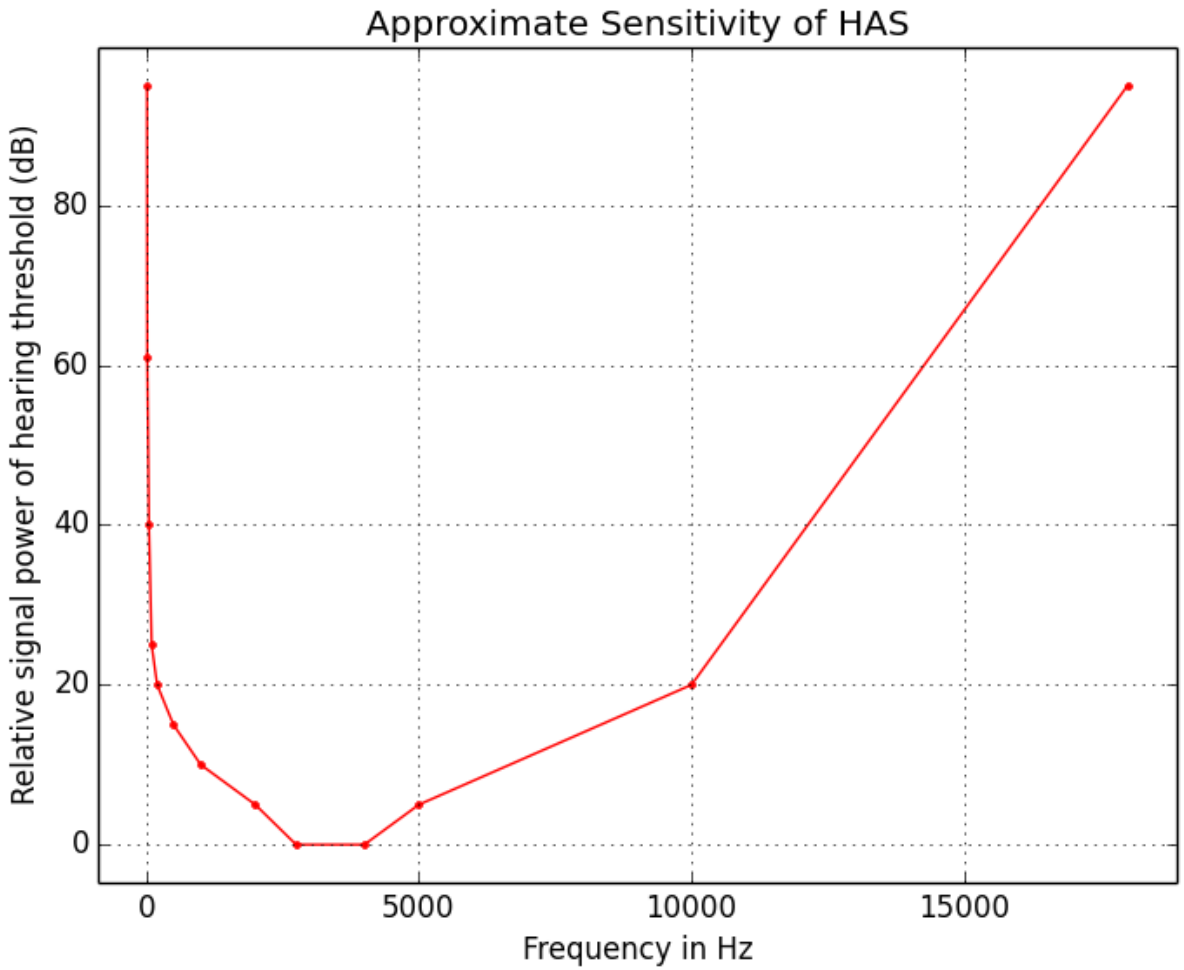
- **Sound waves enter the ear and travel through the ear canal**
- **Eardrum vibrates**
- **Eardrum vibrations cause small bones to vibrate (hammer, anvil, and stirrup)**
- **Bone vibrations cause wave patterns in fluid inside cochlea**
- **Cochlea is lined with a membrane containing hairs. The membrane's stiffness varies along the cochlea**
  - Different parts of cochlea respond to different frequencies
  - Mechanical Fourier Transform

Cochlea


These hairs respond to high frequencies
These hairs respond to low frequencies

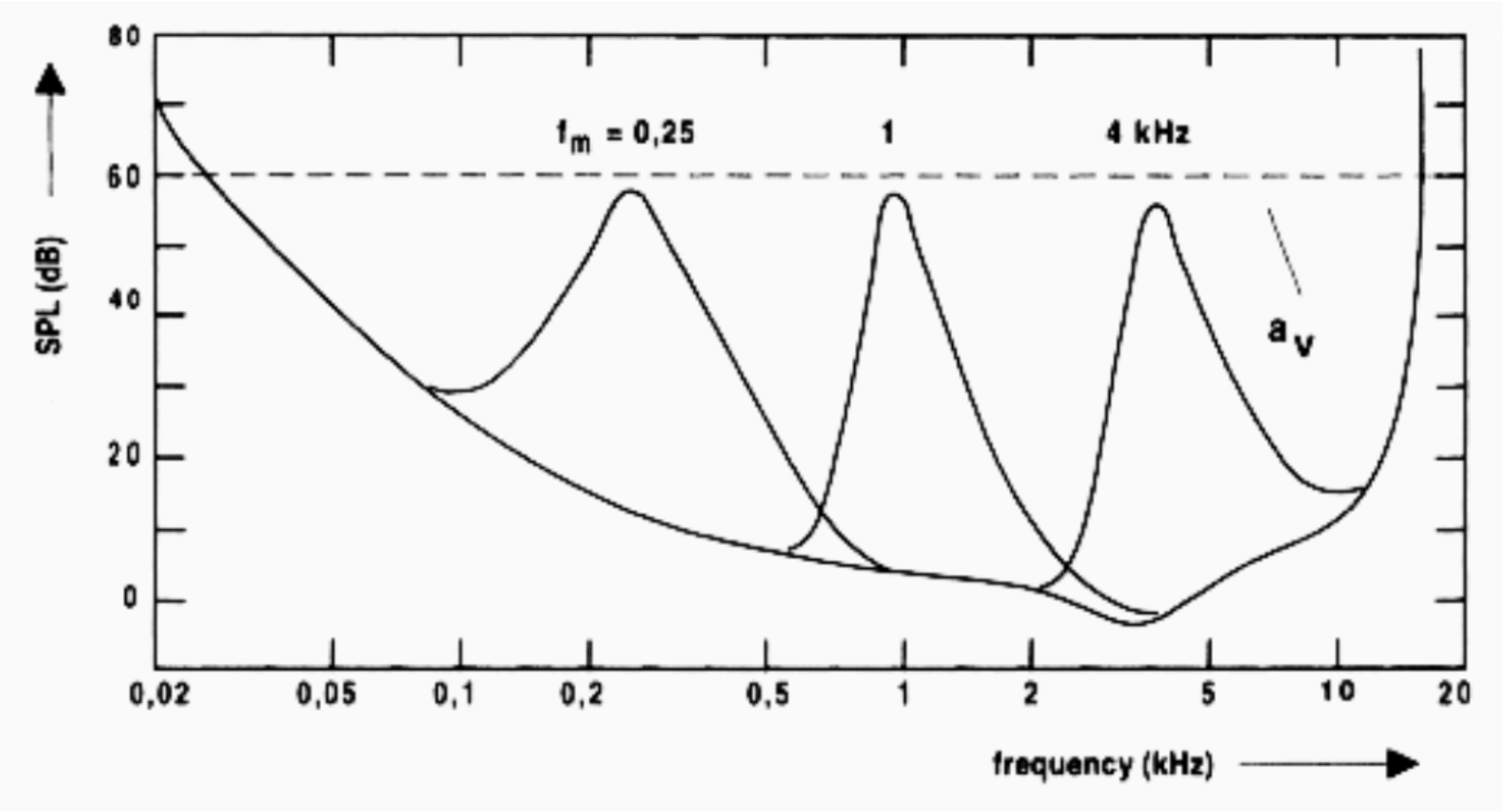
Threshold of Hearing for Human Auditory System

CU Boulder



Audio Masking (Critical Bands)

CU Boulder



Some Sound Terminology

- **Loudness**
  - The quality that allows us to judge one sound as quieter than another
- **Pitch**
  - The quality that allows us to judge one sound as “higher” than another
- **Timbre (tone color, tone quality)**
  - The quality that allows us to judge two sounds as different even if they have the same ”pitch” and “loudness” (think of two instruments playing the same note)

Physics vs Perception

- **Human perception does not linearly relate to physically measurable entities**
  - Intensity: a physically measurable entity (related to loudness)
  - frequency: a physically measurable entity (related to pitch)
- **Example: twice the physical intensity is not perceived as twice the loudness (hearing) or brightness (vision)**
- **Example: twice the physical frequency does not sound twice as high to the ear (especially at higher frequencies)**