Digital Audio and Speech Processing
(Sayısal Ses ve Konuşma İşleme)

Prof. Dr. Nizamettin AYDIN

naydin@yildiz.edu.tr
nizamettinaydin@gmail.com
http://www3.yildiz.edu.tr/~naydin

Basic Audio Processing

1

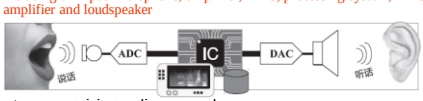
Tools for speech and audio processing

- SoX – Sound eXchange (free)
  - a command-line audio processing tool
  - https://sourceforge.net/projects/sox/
- Audacity (free)
  - an interactive, graphical audio editor
  - https://www.audacityteam.org/
- Matlab (notfree)
  - https://www.mathworks.com/
- GNU Octave (free)
  - https://www.gnu.org/software/octave/index
- Scilab (free)
  - https://www.scilab.org/
- VOICEBOX (Matlab toolbox) (free)
  - http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html
- …

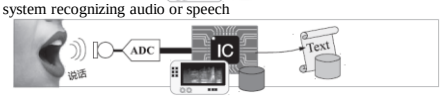
2

Three classes of digital audio system

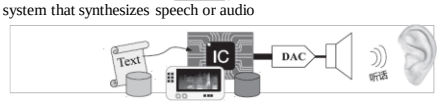
- A complete digital audio processing path
  - including an input microphone, amplifier, ADC, processing system, DAC, amplifier and loudspeaker



- A system recognizing audio or speech



- A system that synthesizes speech or audio



3

Basic audio processing using Matlab/Octave

- Recorded speech or other sounds are stored within MATLAB as a vector of samples,
  - with each individual value being a double precision floating point number.
- A sampled sound can be completely specified by the vector of these numbers as long as one other item of information is known:
  - the sample rate at which the data was recorded.
- To replay the sampled sound, it is only necessary to sequentially output a voltage proportional to the stored vector information,
  - with a gap between samples equivalent to the inverse of the sample rate.

4

Basic audio processing using Matlab/Octave

- Any operation that MATLAB can perform on a general vector can be performed on stored audio
- The audio vector can be loaded and saved in the same way as any other MATLAB variable
- The audio vector can be processed, added, plotted, inverted, transformed and so on
- However, there are some special considerations when dealing with audio

5

Recording Sound

- A sound can be recorded by using
  - a separate audio application to record sound (such as Audacity)
    - then can be exported to a file in .wav format (or any other standard format), which can then easily be read into MATLAB
  - the audiorecorder() function in MATLAB
    - requires an audio recorder object to first be created, specifying sample rate, sample precision in bits, and number of channels, before recording can begin.
  - This is done as follows:

```
ses = audiorecorder(16000,16,1);
record(ses);
```

6

Recording Sound

- After entering the record() command, the computer records sound until stop() command entered.
- Other related commands:
  - pause(10); %waits 10 seconds before next command
  - pause(ses); %this pauses the recording
  - resume(ses); %this resumes recording
- The resulting audio file can be played back as follows:
  - play(ses);

7

Recording Sound

- The ses object is not numerical data;
  - it is a structure that contains the numerical sample data along with various other items of information about the recording.
- To convert the stored recording within ses into the vector of audio samples, use the getaudiodata() function as follows:
  - konusma = getaudiodata(ses, 'double');
- where the resulting data type needs to be specified
- The entire recording and playback sequences operate as background commands, making them a good choice when building interactive speech tools, or using MATLAB for conducting real-time experiments.

8

Storing and replaying audio

- Replaying a numerical vector of sound stored in floating point format:
  - sound(konusma, 8000);
- In MATLAB, to find out what data is stored internally is whos.
  - It gives a list of stored data, indicating the data type and number of items.
- Anything stored inside the memory, and shown in this list can be wiped using clear
- A quick way to remove some stored array data from memory is by overwriting the array with an empty one, indicated by square brackets,
  - For example, ses = [];

9

Storing and replaying audio

- to automatically scale the output amplitude based upon the maximum amplitude element in the audio vector:
  - soundsc(konusma, 8000);
- Similar effect could be accomplished by scaling the audio vector before playing back with the unscaled command sound():
  - sound(konusma/max(abs(konusma)), 8000);
- Plotting a time-domain waveform
  - plot(konusma);
  - plot([1: size(konusma)] / 8000, konusma);

10

Audio file handling

- Wave file (.wav) contains header information that specifies useful attributes like the sampling rate, number of channels and number of bits in each sample
- PCM (pulse coded modulation) and RAW hold streams of pulse coded modulation data with no headers or gaps.
- RAW Audio is an audio file format for storing uncompressed audio in raw form
  - Raw files can have a wide range of file extensions, common ones being .raw, .pcm , .dat, or .sam
- A-law and μ-law are logarithmically compressed audio samples in byte format
  - Each byte represents something like 12 bits in equivalent linear PCM format.
    - This is commonly used in telecommunications where the sample rate is 8 kHz
- MP3, represented by the file extension .mp3, is a standard compressed file format invented by the Fraunhofer Institute in Germany

11

Audio file handling

- Most general audio files are stored in Wave file format (.wav) or in MP3 format
- To load a Wave file the wavread() or audioread() commands can be used.
  - wavread() has been marked to be removed in a future release of MATLAB

```
[y,fs] = audioread('myfile.wav');
```

- The vector y contain the audio data, fs will specify the sample rate.
  - If the file contained stereo data, then y will be a 2-by-*n* matrix

12

Audio file handling

- To read raw PCM sound into MATLAB, use the general purpose fread() function
  - First, open the file to be read by specifying its name (and path, if not in the current directory):
    ```
    fid=fopen('kayit.pcm', 'r');
    ```
  - Then read the entire contents into a vector:
    ```
    konusma=fread(fid , inf , 'int16' , 0, 'ieee-le');
    ```
    - reads in the entire file (infinite values) of 16-bit integers, the format is IEEE little endian, which is the format typically used by a PC.
  - The following would read in an entire file of unsigned 16-bit integers, in big endian format (UNIX).
    ```
    speech=fread(fid , inf , 'uint16' , 0, 'ieee-be');
    ```
  - Close any file opened, after finished reading or writing:
    ```
    fclose(fid);
    ```


Audio file handling

- Use fwrite() to save data
- An array of audio samples named y, with sample frequency fs, can be saved using either the wavwrite() or audiowrite() commands
  ```
  audiowrite('myfile.wav',y,fs);
  ```
- To save and load general arrays of numbers within MATLAB, use save and load commands:
  ```
  save myspeech.mat speech1 speech2
  ```
- The saved arrays can be reloaded into another session of MATLAB by issuing the command:
  ```
  load myspeech.mat
  ```