CS578: Project 2: Project Sinusoidal Model

October 10th 2025

Delivery: November 15th 2025

Questions: yannis@csd.uoc.gr, kafentz@csd.uoc.gr, and/or hy578-list@csd.uoc.gr

During this project you will explore the Sinusoidal Representation of speech signals and you will work with an implementation in Python of the Sinusoidal Model (SM) suggested by McAulay and Quatieri. In the provided Python code, there are some empty command lines that are waiting for you to fill in. Once you do this, you can play with the code to perform speech analysis and synthesis based on SM.

In this project you will use the code in the Python file: `SinM_test_hy578.py`. You will play with a speech signal in a WAV format named `arctic_bdl1_snd_norm.wav`. You can also use files from Project 1 if you like.

Specifically:

1. Analysis-Synthesis of speech based on the Sinusoidal Model

   Download the Python code `SinM_test_hy578.py`.

   Locate the areas with the following message: INSERT CODE HERE

   - **Lines 111-114:**

     Having computed the magnitude spectrum of a speech frame (line 96) using zero-phase analysis (lines 92-95) and normalized window (line 89), you need to perform peak picking on the magnitude spectrum to estimate the frequencies, and the corresponding magnitudes and then compute the corresponding phase information from the phase spectrum, in order to minimize the mean squared error between the input signal and the model (known also as maximum likelihood solution).

   - **Lines 190-192:**

     You must interpolate the estimated and matched (after applying the matching process - see below) amplitudes and phases. For amplitudes you will use linear interpolation while for phases you will use *cubic* interpolation.

Then, once you have the values for the amplitudes and phases for each speech sample, you will perform synthesis by just adding all the components.

- **Lines 220-222, and 234-236:**

  In each frame you compute frequencies, amplitudes and phases. Based on frequency information, you must perform the matching algorithm explained during the lectures. The Δ design parameter for frequency matching was set to 10 Hz (line 197).

  In some cases you will have births and deaths of components.

Once done, you must be able to load the WAV file, perform analysis and synthesis frame-by-frame. At the end you can save your computed speech signal as well the Signal to Noise Ratio (SNR) frame-by-frame. Listen to the original and the processed speech signal.

Write down very briefly your listening impressions.

2. Give us your voice

   Record a speech signal using your voice and perform analysis and synthesis using the Sinusoidal Model. You may use 16000 as sampling frequency during your recording and 16 bits resolution.

   Save both signals and give them to us.

Answers may be given in Greek or in English. Return the functions you wrote by yourself plus the original (initial) Python file with the requested lines filled in.