

Assignments 5 and 6

Mathematical Tools for ITS (11MAI)

Mathematical tools, 2020

Jan Příkryl

11MAI, lecture 12

Monday, November 30, 2020

version: 2021-12-08 12:02

Department of Applied Mathematics, CTU FTS

Assignment 5

Assignment 6

Assignment 5

Use your knowledge of harmonic signal generation and windowing in Matlab to write a script that generates a `.wav` file playing eight tones of the selected music scale.

How do I generate tone frequencies?

- Every octave has 12 half-tones, 8 main tones
- Every octave doubles the frequency, i.e. $f(C_5) = 2f(C_4)$
- Half-tone frequencies form a **geometric series**

| tone | C_4 | D_4 | E_4 | F_4 | G_4 | A_4 | B_4 | C_5 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|
| f [Hz] | 261.63 | 293.66 | 329.63 | 349.23 | 392.00 | 440.00 | 493.88 | 523.25 |

```
% Define note indexes for C major and G major
```

```
c4=40; cis4=41; d4=42; dis4=43; e4=44; f4=45; fis4=46; g4=47; gis4=48;  
a4=49; ais4=50; h4=51; c5=52; cis5=53; d5=54; dis5=55; e5=56; f5=57;  
fis5=58; g5=59;
```

```
% Octaves
```

```
c_maj = [ c4 d4 e4 f4 g4 a4 h4 c5 ];  
g_maj = [ g4 a4 h4 c5 d5 e5 fis5 g5 ];
```

Example (Frequency of C₄ and C₅ based on A₄)

The tone C₄ is **nine half-tones below** A₄, and the tone C₅ is **three half-tones above** A₄, therefore

$$f(C_4) = f(A_4) \cdot 2^{(-9/12)} = 440 \cdot 2^{-0.75} = 261.63 \text{ Hz}$$

$$f(C_5) = f(A_4) \cdot 2^{(3/12)} = 440 \cdot 2^{0.25} = 523.25 \text{ Hz}$$

Using the `c_maj` defined above, generate one octave of C-major scale:

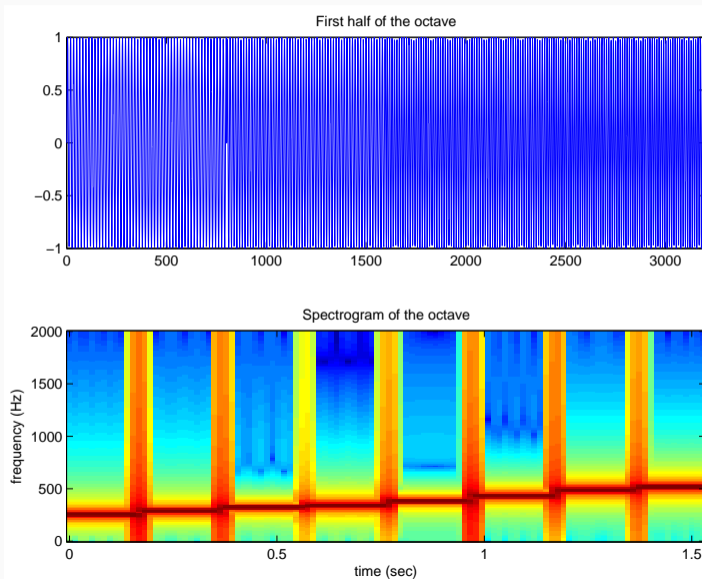
```
Fs = 8000; % sampling frequency
t0 = 0.2; % tone duration
tt = 0:(1/Fs):t0; % sample times
yc = []; % output
for key = c_maj % go through all keys in c_maj
    % compute the tone frequency, freq(a4)=440 Hz
    freq = 440 * (2 .^((key - a4)/12));
    % concatenate the new tone to the previous ones
    yc = [ yc, sin(2*pi*freq*tt) ]; % or cos(2*pi*freq*tt - phi)
end
sound(yc, Fs); % play it with the given sampling frequency
```

The tone transitions are quite disturbing, let's check with spectrogram:

```
Fmax = Fs/4; % maximum frequency
Nfft = 256; % number of FFT coefficients
Nover = 200; % number of overlapping samples
% Display the spectrogram
figure(1);
spectrogram(yc, Nfft, Fmax, [], Nover);
title('Spectrogram of the C-major without windowing');
```

The distortion (“clicking”) occurring at tone transition point is due to an abrupt change of the waveform amplitude. This can be mitigated e.g. by applying a window of suitable length to the tone sample.

Assignment 5 — C-major spectrogram with transients



- a) Repeat the steps outlined above for G-major scale, with samples stored in `yg` vector.
- b) Save your sound using `audiowrite('g_major_orig', yg, Fs)`.
- c) It sounds rather artificially, can you find the way to improve the generated record?
- d) **Bonus:** Repeat the steps (a) and (b) for windowed tone samples, write the output to `g_major_window.wav`.
- e) Deliver the code that generates the `.wav` file(s) as a ZIP file by December 23, 2021. No report is necessary.

- a) Repeat the steps outlined above for G-major scale, with samples stored in `yg` vector.
- b) Save your sound using `audiowrite('g_major_orig', yg, Fs)`.
- c) It sounds rather artificially, can you find the way to improve the generated record?
Hint: Use a window function to overlap the tones!
- d) **Bonus:** Repeat the steps (a) and (b) for windowed tone samples, write the output to `g_major_window.wav`.
- e) Deliver the code that generates the `.wav` file(s) as a ZIP file by December 23, 2021. No report is necessary.

- a) Repeat the steps outlined above for G-major scale, with samples stored in `yg` vector.
- b) Save your sound using `audiowrite('g_major_orig', yg, Fs)`.
- c) It sounds rather artificially, can you find the way to improve the generated record?
Hint: Use a window function to overlap the tones! **Hint:** Add harmonics!
- d) **Bonus:** Repeat the steps (a) and (b) for windowed tone samples, write the output to `g_major_window.wav`.
- e) Deliver the code that generates the `.wav` file(s) as a ZIP file by December 23, 2021. No report is necessary.

- a) Repeat the steps outlined above for G-major scale, with samples stored in `yg` vector.
- b) Save your sound using `audiowrite('g_major_orig', yg, Fs)`.
- c) It sounds rather artificially, can you find the way to improve the generated record?
Hint: Use a window function to overlap the tones! **Hint:** Add harmonics! If you want to be even cooler, look at **additive**, **subtractive** or **FM** synthesis.
- d) **Bonus:** Repeat the steps (a) and (b) for windowed tone samples, write the output to `g_major_window.wav`.
- e) Deliver the code that generates the `.wav` file(s) as a ZIP file by December 23, 2021. No report is necessary.

- a) Repeat the steps outlined above for G-major scale, with samples stored in `yg` vector.
- b) Save your sound using `audiowrite('g_major_orig', yg, Fs)`.
- c) It sounds rather artificially, can you find the way to improve the generated record?
Hint: Use a window function to overlap the tones! **Hint:** Add harmonics! If you want to be even cooler, look at **additive**, **subtractive** or **FM** synthesis.
- d) **Bonus:** Repeat the steps (a) and (b) for windowed tone samples, write the output to `g_major_window.wav`.
- e) Deliver the code that generates the `.wav` file(s) as a ZIP file by December 23, 2021. No report is necessary.

Assignment 5

Assignment 6

Assignment 6

- a) Extend the approach from Assignment 5 to take into account the length of the tones — probably the best solution would be to use an array of tone lengths:

```
jednadve_t = [a4 a4 f4 a4 a4 f4 a4 a4 h4 a4 a4 g4 ]; % tones  
jednadve_d = [ 8 8 4 8 8 4 8 8 8 8 4 4 ]; % tone lengths 1/8 and 1/4 of t0
```

and modify the generator of sample times `tt` to end at `t0/tdiv`:

```
for i = 1:length(jednadve_t) % go through all keys indexes  
    key = jednadve_t(i); % tone number  
    tdiv = jednadve_d(i); % length divisor for the tone  
    tt = 0:(1/Fs):(t0/tdiv); % t0 is now the bar length  
    freq = 440*(2.^((key-a4)/12));  
    sample = sin(2*pi*tt); % can be modified by windowing  
    y = ...  
end
```


- b) Select a Christmas Carol *that has not been composed in C-major scale*.
- c) Compose the Carol using Matlab commands.
Generate the sampled tones using the non-windowed approach from Assignment 5 taking into account different tone lengths as suggested in (a) of this list. Save the result to a **.wav** file.
- d) **Bonus:** Compose a Christmas Carol with a tone scale improved by windowing, harmonics or some other approach mentioned in the bonus part of Assignment 5.
- e) Deliver the code that generates the **.wav** file as a ZIP file by December 23, 2021. No report is necessary.