

Project 2: E-Music

Due: Wednesday, May 2, 2007, 10:25am (end of class)

Remarks:

- The *Music analysis* part 3(a) is the heart of this project. So, be sure you allocate enough time for this task. The figures of merit are the correctness of the analysis and the parameter L .
- Project groups:
 - You are encouraged to work in teams of two, submitting one joint report and getting the same grade.
 - For those who decide to do the project alone, this will be taken into account when grading.
 - Please tell me by April 6 whether you are working alone or in a team. In the latter case, include the name of your teammate.
- Files:
 - All files can be found at `/afs/nd.edu/course/sp.07/ee/ee40471.01/project2/` and at `http://www.nd.edu/~jkliewer/EE40471/project2/`.
 - All signals are sampled at 8192 Hz. Use the same sampling frequency for the signals you generate.
 - Files with the extension `{au,wav}` can be loaded into Matlab using `{au,wav}read`.
- Report:
 - Follow the guidelines in `guidelines.ps` (handed out with the first project).
 - Make your Matlab files available at `/afs/nd.edu/course/sp.07/ee/ee40471.01/dropbox/(your_net_id)/project2/`.

1 Introduction

One of the many applications of digital signal processing is the manipulation of musical signals. Music signal processing includes a wide variety of effects and filtering techniques. This project deals with the synthesis, analysis, and processing of music.



(a) Notation.



(b) On a keyboard.

Figure 1: C major scale.

Notes. In Western music, the frequencies of musical instruments obey a geometric series formula $f_i = f_{\text{ref}}q^i$, where $q = 2^{1/12}$ and f_{ref} is the reference frequency, usually 440 Hz, which corresponds to the musical note A. The ratio $q = 2^{1/12}$ is called a *semitone*. The range of a piano is $-48 \leq i \leq 39$ (88 keys), corresponding to frequencies between 27.5Hz and 4186Hz. There are 12 semitones in an *octave*, and the corresponding notes are named A, Bb, B, C, C#, D, D#, E, F, F#, G, and G#. Generally, if a 'b'¹ is added to the name of a note, it denotes the next semitone below the original note; if the '#'² sign is added, it refers to the next semitone above the original note. The note 12 semitones above A is again called A and is said to be an octave higher, hence the ratio between two notes an octave apart is exactly 2. The 7 natural notes A, B, C, D, E, F, G are played by the white keys of the piano, the sharp and flat notes by the black keys.

Scales. Western music is largely based on *diatonic* scales. A diatonic scales consists of 7 notes per octave. The diatonic C major scale consists of the natural notes C, D, E, F, G, A, B, and their octaves (Fig. 1). The frequency of the “middle” C is $f_C = 440 \cdot 2^{-9/12} \approx 262$ Hz; the frequencies of the notes in the C major scale are $f_j = f_C q^j$ with $j \in \{0, 2, 4, 5, 7, 9, 11\}$. In total, there a 12 major keys, each starting at a different semitone. In addition, there are 3 kinds of minor keys: natural, harmonic, and melodic. In this project, we focus on the former two. The natural A minor scale consists of the natural notes A, B, C, D, E, F, G. In the harmonic A minor scale, the G is replaced by G#. As for the major scales, there are 12 natural and 12 harmonic minor scales. Hence, if f_0 is the frequency of the root (the base note), the notes in the minor scale have the frequencies $f_j = f_0 q^j$ with $j \in \{0, 2, 3, 5, 7, 8, 10\}$ for the natural minor scale and $j \in \{0, 2, 3, 5, 7, 8, 11\}$ for the harmonic minor scale. As examples, one octave of each the C major, natural and harmonic minor scales is included in the file `c_scales.mat`. They can be played using the `sound` or `soundsc` commands.

Chords. A *chord* is a group of notes played simultaneously. In theory these can be any notes, but different combinations produce different musical qualities, and so in practice the notes used will usually be based on a set of guidelines governing the formation and use of chords. The study and use of chords is sometimes referred to as harmony.

A *triad* is a simultaneous combination of three notes and is the simplest form of chord. The three notes are the first, third, and fifth note in a scale. In the case of a major scale, these are the notes $f_k = f_0 q^k$ with $k = 0, 4, 7$; in the case of a minor scale, it's $k = 0, 3, 7$. Triads may also be inverted. An inversion is where the root note is not the lowest in the triad.

For example, the C major triad (or chord) consists of C, E, and G. `chords.html` visualizes the C, F, and G major triads with their inversions by the respective keys on the piano. The chords are named after the lowest note in the root position.

¹Spoken: “flat”, i.e. 'Bb' is read as “B-flat”.

²Spoken: “sharp”, i.e. 'F#' is read as “F-sharp”.

Harmonics. The harmonic structure of a musical instrument is the series of amplitudes of the various harmonics relative to the fundamental frequency. The harmonic structure depends mostly on the type of the instrument (e.g., bow instrument, plucked string instrument, wind instrument), but also on the note played and the way it is played. The most important harmonics are the second (one octave above the fundamental frequency) and third harmonics. Some examples of spectra of musical instruments are given in `harmonics.ps`. It can be seen that the amplitudes of the even and odd harmonics are characteristic for different instruments.

2 Music Synthesis

In this section, you will develop a Matlab program that synthesizes “realistic” e-music. Every piece of music consists only of the notes of one scale.

Write a Matlab function `basic.m` that randomly generates a piece of e-music. The function takes four inputs: the length of the piece P (in seconds), the duration of a full note L (in seconds), the key (major, natural, harmonic), and the root note to define the scale. The output is the basic music signal. The first note in the piece must be the root note (length L). Of course, the piece may extend over several octaves. As in “real” music, the notes may differ in duration, but the length of the notes is restricted to the set of full notes, half notes, and quarter notes, i.e., notes of duration $\{L, L/2, L/4\}$. So, the number of notes in the piece is between P/L and $4P/L$. Choose a constant amplitude of one for all notes.

For further reference, we denote the signals generated by this program as *basic music*.

3 Music Analysis

Assume now you are given a music signal. The task in this section is to reconstruct the music, note by note.

- (a) Write a Matlab function `analysis.m` that takes a *basic music signal* as its input and lists the notes (by their names) and their duration (full, half, quarter). Remember that the first note in the basic piece is the root note. Your program also has to determine the scale and key of the music signal (major, natural minor, harmonic minor). The performance of your program will be tested using basic music signals of different speeds L . The minimum L at which your analysis program works without error is the performance measure. Determine this minimum L for your algorithm.
- (b) What is the performance of your program for non-basic music (i.e., music that includes harmonics, reverberations, and/or chords)? Beside using signals you produced yourself, use recorded music such as `twinkle.wav`, `bugle.au`, `bach.organ.au`, `beet5.wav`, `for_elise.wav`, `gmin_np.wav`. A challenging example is `etude.au`. The performance is probably not satisfactory: How could you improve your analysis program for these cases? Explain your suggested improvements (no code required here).

4 Music Processing

Combining the results from the previous sections, you are now developing software that processes basic music signals. That is, write programs that:

- (a) Add triads. Write a function `add_triads.m` that takes a basic music signal as its input and outputs a music signal with triads.

- (b) Noise removal filtering of a noisy piece.

The file `c_noisy.mat` contains the variables `c_noisy_m10`, `c_noisy_0`, and `c_noisy_10` containing three short pieces in C major with SNRs of -10 , 0 , and 10 dB, respectively (the SNR was defined in Project 1). Knowing that the noise is white (i.e., spectrally flat), that the piece is in C major and consists of only 8 notes, design a set of filters that remove as much of the noise as possible without causing audible distortion. Implement the filters in a function `noise_removal.m` that takes the (noisy) signal as its input and outputs the clean signal. Determine the SNRs after the filtering.

The short pieces contain the following notes: $k=[1\ 2\ 3\ 4\ 5\ 6\ 6\ 5\ 3\ 5\ 3\ 5\ 7\ 7\ 8]$ (1=C, 2=D, etc.) of durations $\text{dur}=0.15\text{s}\cdot[2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 2]$.

- (c) How does your analysis function from part 3(a) perform on those noisy signals? Rather than using filters to remove the noise, is it better to analyze the noisy signal to extract the notes, and then re-synthesize the piece noiselessly? What are the SNR values that you get with this method?
- (d) Do something else which is exciting and/or useful.