ECE 417 Lab Exercise # 3
Music synthesis

In this lab, you will work on Matlab-based exercises designed to familiarize you with the task of synthesizing musical notes using digital signal processing and combining the notes to synthesize simple tunes. You will learn to generate an impulse train with the right pitch period which will then be applied to FIR and IIR linear shift-invariant systems to generate the sounds for a specified musical note. Several musical notes will be suitably combined to create simple tunes.

You will learn to determine the pitch period for a given note frequency and create a train of unit impulses for a desired note duration. The impulse train is applied as excitation to a system with Impulse response $h[n]$ and a transfer function $H(z)$. The output amplitude will be shaped to produce pleasant sounding note. Several notes so created with desired duration will be concatenated to produce a simple tune.

We will assume that the sampling rate f_s of the signal is 44100 Hz. We will use the pitch frequencies given in the table below:

Table 1: **Notes and picth frequencies**

Note	Pitch frequency
C_4	**261.61**
$C_4^{\#}/D_4^b$	**277.18**
D_4	**293.66**
$D_4^{\#}/E_4^b$	**311.13**
E_4	**329.63**
F_4	**349.23**
$F_4^{\#}/G_4^b$	**369.99**
G_4	**392.00**
$G_4^{\#}/A_4^b$	**415.30**
A_4	**440.00**
$A_4^{\#}/B_4^b$	**466.16**
B_4	**493.88**
C_5	**523.25**

Values extracted from longer table at:
https://pages.mtu.edu/ suits/notefreqs.html

1 Violin sound synthesis applying impulse train excitation to a finite-duration impulse response (FIR) filter

1.1 Creating a single note

We will use a pitch frequency $f_{pitch} = 1/T_{pitch}$ where T_{pitch} is the pitch period. With $f_s = 44100$ samples/sec, the pitch period will have $N_p = f_s T_{pitch} = f_s/f_{pitch}$ samples with the first sample equal to 1 and the remaining $N_p - 1$ samples zero. Create an impulse train x_{A_3} with impulses N_p samples apart corresponding to a pitch frequency $f_{pitch} = f_{A_3} = 220$ Hz. Use rounding to get an integer value N_p. The impulse train signal duration should be 1 second.

Apply the impulse train to an FIR filter with impulse response $h[n]$ given by the vector 'impresp_violin' stored in the file irviolin.mat, which can be accessed with the command

```
load('irviolin.mat');
```

Matlab command "filter" is convenient to implement the FIR filter.

The waveform envelope of the output signal y_{A_3} should be roughly constant. The envelope can be shaped to have a more gradual rise and fall. Shape (or modulate) the amplitude by multiplying the signal with a suitable scaling function. One option to do this is scale the signal with the square of half the period of a sine wave of appropriate duration. Call the shaped signal v_{A_3}.

Some sample code:

```
load('irviolin.mat'); % loads the vector impresp_violin
a = 1; % denominator for FIR filter
b = impresp_violin;
Fs = 44100; % Sampling frequency, # of samples in 1 sec, integer
Tdur = 2; % The note should be played for 2 seconds
Ndur = round(Tdur*Fs); % # of samples in Tdur seconds
n=[0:Ndur-1]; % index set [0 1 2 3 ... Ndur-1]
scalefn = sin(pi*n/(Ndur));
scalefn = scalefn.*scalefn; % modulating function for the note
Fpd = 220; % pitch frequency desired

Np = round(Fs/Fpd); % # samples in pitch period, rounded.
% Np is used in pulse train generation
tem = n/Np - round(n/Np);
```

```
% entry in tem is zero when corresponding entry in n is a multiple of Np
pt = (tem == 0); % creates pulse train
x = filter(b,a,pt);% generates signal y_{A_3} for note A3 of duration Ndur s
x = x.*scalefn; % generates shaped signal v_{A_3}
note = x;
soundsc(note,Fs);
audiowrite('note.wav',note,Fs)
```

Q1: Play the sound of the output signal y_{A_3} and the shaped signal v_{A_3}. Comment on the difference in the of the sound of the note produced.

1.2 Creating a tune

Now create a sequence of notes with appropriate duration and concatenate the notes to create the following tune (from the song "Doe a deer ...")

C_4 D_4 E_4 C_4 $E_4 - C_4$ E_4

Doe, a deer a female deer...

(See https://noobnotes.net/do-re-mi-sound-of-music/.

Also see http://www.choose-piano-lessons.com/piano-notes.html for other simple tunes of nursery rhymes)

You may use a different tune containing at least 6 notes including notes with at least 3 different pitch frequencies.

Q2. Explain the details of your procedure. Load your audio file to Blackboard with your report.

2 Violin sound synthesis applying impulse train excitation to an infinite-duration impulse response (IIR) filter

2.1 Creating a single note

Repeat the procedure as in section 1 to create an impulse train x_{A_3} with impulses N_p samples apart corresponding to a pitch frequency $f_{pitch} = f_{A_3} =$ 220 Hz.

Apply the impulse train to an IIR filter with transfer function $H(z)$ specified by the numerator coefficient 0.1 and denominator polynomial with coefficients:

$$A = [1.0000\ -2.4584\ 1.8539\ -0.2387\ -0.0466\ -0.1614\ -0.1004\ 0.1980\ -0.0420];$$

Matlab command “filter” is convenient to implement the IIR filter.

The waveform envelope of the output signal y_{A_3} should be roughly constant. As before, use amplitude modulation appropriate for a pleasant sound. Again call the shaped signal v_{A_3}.

Q3: Plot and compare the waveform for 3 pitch periods of the output signal y_{A_3} in the cases of synthesis by FIR and IIR filters. Comment on the difference in the sound of the note produced by the two methods.

2.2 Creating a tune

As before create a sequence of notes with appropriate duration and concatenate the notes to create the same tune you created in the case of FIR filter music synthesis.

Q4. Explain the details of your procedure. Load your audio file to Blackboard with your report.

Q5: Comment on the difference in the of the sound of the tunes produced by synthesis using FIR and IIR filters.

Written by R. Ansari. Updated Fall 2022.