

ENGIN 100: Music Signal Processing

PROJECT #1: Tone Synthesizer/Transcriber

Professor Andrew E. Yagle

Dept. of EECS, The University of Michigan, Ann Arbor, MI 48109-2122

I. ABSTRACT

This project teaches you the Matlab Graphical User Interface (GUI) tools you will be using in the other two projects. It also demonstrates some issues you will face during the projects. The goals of this project are: (1) To write a Matlab program for a tone synthesizer with an on-screen mouse-activated keypad; (2) To write a Matlab program for a tone transcriber that produces a Matlab stem plot similar to musical staff notation from a pure tonal signal. The second project will do the same thing for telephone touch-tones; the third project will do the same thing for synthesized musical instruments. This project will help you do both.

II. BACKGROUND

In Lab #2 you learned the frequencies used in a pure tonal version of “The Victors.” In this lab you will apply that knowledge to build a tone synthesizer and transcriber. You need to know some musical notation and nomenclature, and some graphical Matlab commands. These are covered in the next two sections.

III. BASICS OF MUSICAL STAFF NOTATION

A. Results from Lab #2

In Lab #2 you learned that the musical tones used in “The Victors” had the following frequencies: 392, 440, 494, 523, 587, 659 Hertz (rounded to the nearest integer). You also inferred that there were “missing” frequencies: 415, 466, 554, 622 Hertz. You also deduced that these notes are related to each other not by common differences, but by a common ratio of 1.06 (actually 1.0595).

In fact 1.0595 is the 12th root of 2 ($2^{1/12}$). That immediately suggests that there is a basic set of 12 notes, which “repeat” with their frequencies multiplied by integer powers (positive and negative) of two. This is indeed the case. This is the modern *12-tone* or *chromatic* musical scale; most Western music is based on it.

The repetitions of the 12 frequencies are called *octaves*, since a span of 8 whole tones doubles the frequency. So 880 Hertz is one octave above 440 Hertz, and 220 Hertz is one octave below 440 Hertz. We will focus on the octave spanned by the frequencies in “The Victors,” since many musical compositions use this octave.

However, the 88 keys of a piano keyboard span 8 octaves:

- Rightmost key: 4186 Hertz is 3 octaves above 523 Hertz ($4186/2^3 \approx 523$)
- Leftmost key: 27.5 Hertz is 4 octaves below 440 Hertz ($440/2^4 \approx 27.5$).

B. MIDI Frequency Notation

This information is enough to create a notation for most Western music, The Musical Instrument Digital Interface (MIDI) notation represents musical frequencies with integers from 0 to 127 using the formula

$$MIDI = 69 + 12 \log_2(F/440)$$

where F is frequency in Hertz. In MIDI notation, frequencies of “The Victors” are represented as

71,67,69,71,67,69,71,72,69,71,72,69,71,72,74,76,71,71,72,67,69,71,74,71,69,67

MIDI also includes information about amplitude, duration, and some other things.

C. Musical Staff Notation

Musical staff notation (the musical notation you usually see) is more complicated. The reason is that the frequencies used in “The Victors” are much more commonly used in Western music than the “missing” frequencies from Lab #2 (that’s why they were missing).

The 7 tones in “The Victors” are called *whole tones* and designated with letters: A, B, C, D, E, F, G.

The 5 “missing” tones are called *accidentals* or *sharps* and *flats*, and they are designated by “#” and “b”

A#=Bb, C#=Db, D#=Eb, F#=Gb, G#=Ab (pronounced “A-sharp” and “B-flat,” etc.).

Bb is below B (between A and B) and C# is above C (between C and D); A# is the same as Bb, etc. The entire group of 12 tones (which can be represented using either listing below)

{A, A#, B, C, C#, D, D#, E, F, F#, G, G#} same as {A, Bb, B, C, Db, D, Eb, E, F, Gb, G, Ab}

are called *semitones* because they include both *whole tones* and their halfway-between *accidentals*. On a piano keyboard, the whole tones are the white keys and the accidentals (sharps and flats) are the black keys.

Why are the whole tones used much more than the accidentals? Because they sound better! The reason they sound better is that the ratios between their frequencies are very close to ratios of small integers. For example, A is 440 Hertz and E is 659 Hertz, almost exactly a 3:2 ratio. Indeed, Western music was first based on using these ratios of small integers, rather than the equally-spaced logarithms of frequencies used today. The change occurred when J.S. Bach composed “The Well-Tempered Clavicord” using the 12-tone scale. It sounded better than using the frequencies based on small integer ratios (“The Ill-Tempered Clavicord?”).

This can all be summarized in the following table:

NOTES	A	A#	B	C	C#	D	D#	E	F	F#	G	G#
$\frac{\text{FREQS}}{440 \text{ Hz}}$	$2^{0/12}$	$2^{1/12}$	$2^{2/12}$	$2^{3/12}$	$2^{4/12}$	$2^{5/12}$	$2^{6/12}$	$2^{7/12}$	$2^{8/12}$	$2^{9/12}$	$2^{10/12}$	$2^{11/12}$
Hertz	440	466	494	523	554	587	622	659	698	740	784	830
Ratio	1:1	none	9:8	6:5	5:4	4:3	none	3:2	none	5:3	16:9	15:8

The ratios are only approximate; changing from these ratios to a constant ratio of $2^{1/12}$ between notes gave the well-tempered 12-tone scale. Then different keys can be used without affecting frequency ratios.

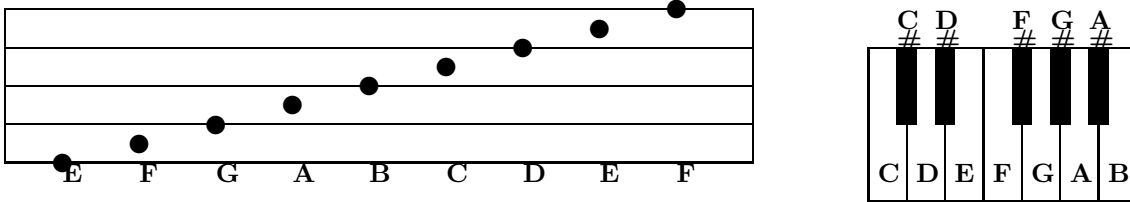
There is no reason to start with A. Starting with G,C and A and looking mostly at whole tones, we have

G	A	B	C	D	E	F#	G	WHOLE	C MAJOR:	C,D,E,F,G,A,B,C
1:1	9:8	5:4	4:3	3:2	5:3	15:8	2:1	NOTES	INTERVAL:	1,1, $\frac{1}{2}$,1,1,1, $\frac{1}{2}$
DO	RE	MI	FA	SO	LA	TI	DO	ONLY	A MINOR:	A,B,C,D,E,F,G,A

Here DO-RE-MI-FA-SO-LA-TI-DO will be familiar to anyone who has seen the movie *The Sound of Music*.

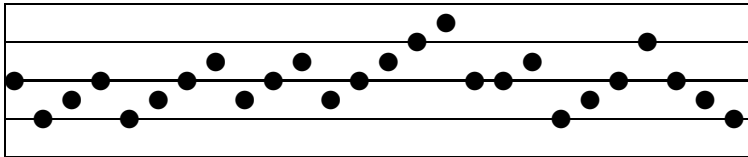
A *musical staff* consists of 5 parallel horizontal lines with circles (notes) on it. Horizontal position is time (read left to right) and vertical position is note frequency. Both the lines themselves, and the spaces between lines, are used to represent **whole notes only**; accidentals (sharps and flats) have a “#” or ”b” in front of the note, respectively. This seems messier than MIDI notation, but since accidental notes occur so less often, it actually simplifies reading the notation while playing an instrument, they tell me (*sight-reading*).

Below left: Musical staff representation of whole notes. Below right: Notes on a piano keyboard.



The thing that looks like a script G in front (not shown) is called a *treble clef*. The small circular lower end encircles the line representing note G; usually this is the second line from the bottom, as shown above (this is the worst notation I’ve ever seen). Lines that look like flag staffs and flags extend from the circles to represent the duration of that note. I won’t get into bass clefs and the rest (including rests) here.

A musical transcription of the chorus of “The Victors,” *without the staves*, then looks like this:



Actual sheet music for the chorus of “The Victors” (obtained from the UM web site) is on the next page.

let the bells them ring, For here they come with banners flying, Here they come, Hur -

Rec. *

rah! Hail to the vic - tors val - iant

melody non legato

mf

without Pedal

Hail! to the con - qu'ring he - roes, Hail! Hail! to

Mich - i - gan the lead - ers and best,

IV. MATLAB GRAPHICS USED IN THIS PROJECT

You will need to use three new Matlab commands in this project. Each one is discussed below.

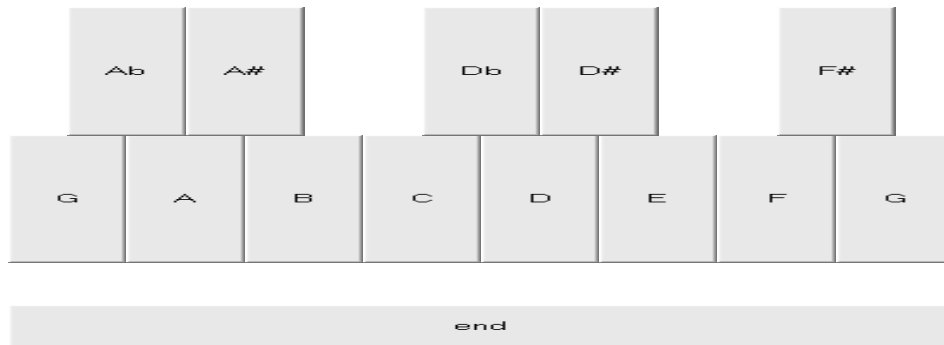
```
>> uicontrol('Style','Pushbutton','Position',[200 400 50 100],'String','A','Callback',
'sound(cos(2*pi*440*[1:3000]/8192))'); (all of this command must be on one line)
```

Creates an on-screen rectangular pushbutton of size 100×50 screen pixels, with lower left corner 200 pixels from the left border and 400 pixels above the bottom border, and puts an 'A' in the middle of it. So the pushbutton corners are at Cartesian coordinates $\{(200, 400), (250, 400), (200, 500), (250, 500)\}$, where the origin is the lower-left corner of the screen, and distance is measured in pixels. When the button is "pushed" by putting the mouse cursor over it and left-clicking, it executes the Matlab command shown. Here, that sounds a 440 Hertz tone for $\frac{3000}{8192}$ seconds. The role of each part of the command should be evident here.

You should be able to use a sequence of commands like this to create an on-screen keyboard, labelled with all of the various notes, and with the accidental notes (sharps and flats) in a row above the whole notes, just like an actual piano keyboard. You can even color the pushbuttons black and white, or you can label them.

The command for each pushbutton must both sound the tone and concatenate it with previous ones:

```
Z=cos(2*pi*440*[1:L]/FS);X=[X Z];sound(Z,FS); for length of L samples and sampling rate FS.
```

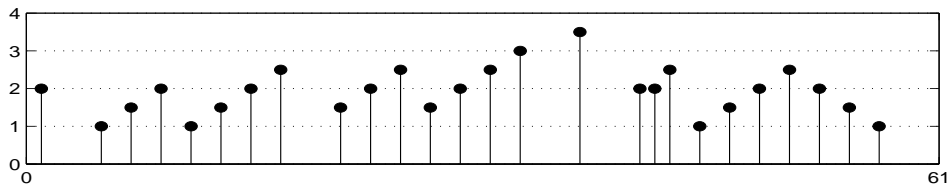


```
>> subplot(311),stem(X,'filled'),axis([0 length(X) 0 4])
```

Produces a stem plot of the numbers in vector X with axes as shown. Try `>> stem([3 1 4 1 5 9])` to see a stem plot. You can fill in the circles using `stem(X,'filled')`, and even change the stem lines.

```
>> set(gca,'YGrid','on','YTick',[0 1 2 3 4])
```

Changes the graph so that it looks like the musical staff notation shown above. If you want solid lines, you can add `'GridLineStyle','-'` to this command. You can also add labels, titles, and other things.



```
>> save proj1.mat X Y and >> load proj1.mat
```

Saves the variables `X` and `Y` to a file `proj1.mat`; recovers all of the variables saved in `proj1.mat`.

V. PROJECT #1: WHAT YOU HAVE TO DO

The results of this project are two `.m` files, one implementing a synthesizer and one implementing a transcriber. You also have to demonstrate to the lab IA that they work, as described below.

Remember throughout that sharps and flats complicate things; MIDI transcription would be easier. This lab would be much easier if they didn't exist, but life isn't like that (engineers must remember this).

A. Musical Tone Synthesizer: On-screen Keyboard

Write a Matlab program (and store it as an `.m` file) that:

- Creates an on-screen keyboard, using a sequence of `uicontrol` commands, that resembles the one octave of a keyboard on which the chorus of “The Victors” is played. Keys should be tall and thin, like a piano;
- The keyboard should run from “C” to “G” and include, in a row above, accidentals in between;
- Produces the appropriate tone when pressed by left-clicking the mouse on it;
- Includes an “end” key that plays back the notes in order, and writes the signal to `proj1.mat`;
- Makes each tone last $\frac{2000}{7999}$ seconds at a sampling rate of $7999 \frac{\text{SAMPLE}}{\text{SECOND}}$;
- Using a sampling rate of $8192 \frac{\text{SAMPLE}}{\text{SECOND}}$ makes some values of the tonal signal zero, so that the frequency estimation algorithm would be dividing by zero! See if you can figure out why;
- Show you can play “The Victors” and “Taps” on it. You may need to practice to get the notes right;
- This program is almost entirely a sequence of `uicontrol` commands, one for each piano key.

B. Musical Tone Transcriber

Write another Matlab program (and store it as an `.m` file) that:

- Loads the file `proj1.mat` generated by your synthesizer above;
- Prints out on the screen a musical transcription of the tones in musical staff notation using Matlab's `stem`;
- Uses the frequency estimation algorithm used in Lab #2, and assumes each tone lasts $\frac{2000}{7999}$ seconds;
- Uses `load('proj1.mat'); Y=reshape(x',2000,length(x)/2000)`; where `x` is your synthesizer output;
- Uses `F=7999/2/pi*acos(mean((Y(3:2000,:)+Y(1:1998,:))./Y(2:1999,:)/2))`; to get frequencies;
- Only has to work for the octave in which the chorus of “The Victors” is played;
- On translating the frequency estimates into notes: Let `K=12*log2(F/440)+6`. Explain the “+6”
- Then `stem S(I)` where `S=[0 .5 .75 1 1.25 1.5 1.75 2 2.5 2.75 3 3.25 3.5 4]`; plots correctly.

C. Project Report

- Your team will present the results of this project in an *oral presentation* to the entire class.
- Send the two `.m` files, named `yourteamname1.m` and `yourteamname2.m`, to your lab instructor.
- Also address these issues: Should you be able to copyright any sequence of tones you can create?
- If we can write tonal signal as a mathematical expression, can music legitimately be copyrighted?