waveform of Fig. 3.16 is *nonperiodic*. We can see why this is so by examining Fig. 3.17, which shows the spectra of the two signals in Figs. 3.16 and 3.15. These *spectrum plots* show "how much" of each cosine wave is in the sum, and they are very similar. However, the frequencies are different: $10\sqrt{8} = 28.28\ldots \approx 30$ and $10\sqrt{27} = 51.96\ldots \approx 50$. These slight shifts of frequency make a dramatic difference in the time waveform.

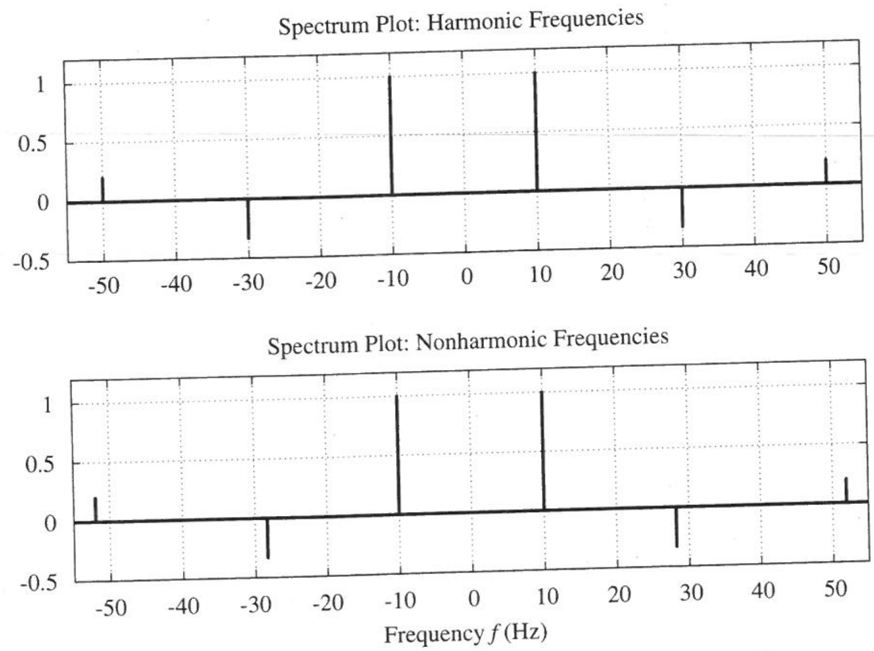

Figure 3.17 Spectrum of harmonic waveform (top) which has period of 1/10 sec, and a nonharmonic waveform (bottom) that is not periodic.

3.5 TIME–FREQUENCY SPECTRUM

We have seen that a wide range of interesting waveforms can be synthesized by the equation

$$x(t) = A_0 + \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \phi_k) \tag{3.5.1}$$

These waveforms range from constants, to cosine signals, to general periodic signals, to complicated-looking signals that are not periodic. One assumption we have made so far is that the amplitudes, phases, and frequencies in (3.5.1) do not change with time. However, most real-world signals exhibit frequency changes over time. Music is the best example. For very short time intervals, the music may have a "constant" spectrum, but over the long term, the frequency content of the music changes dramatically. Indeed, the changing frequency spectrum is the very essence of music. Human speech is another good example. Vowel sounds, if held for a long time, exhibit a "constant" nature because the vocal tract resonates with its characteristic frequency components. However, as we speak different words, the frequency content is continually changing. In any event, most interesting signals can be modeled as a sum of sinusoids if we let the frequencies, amplitudes, and phases vary with time. Therefore, we need a way to describe such time–frequency variations. This leads us to the concept of a time–frequency spectrum or *spectrogram*.

The mathematical concept of a time–frequency spectrum is a sophisticated idea, but the intuitive notion of such a spectrum is supported by common, everyday examples. The best example to cite is musical notation (Fig. 3.18). A musical score specifies how a piece is to be played by giving the notes to be played, the time duration of each note, and the starting time of each. The notation itself is not completely obvious, but the horizontal "axis" in Fig. 3.18 is time, while the vertical axis is frequency. The time duration for each note varies depending on whether it is a whole note, half note, quarter note, eighth, sixteenth, etc. In Fig. 3.18 most of the notes are sixteenth notes, indicating that the piece should be played briskly. If we assign a time duration to a sixteenth note, then all sixteenth notes should have the same duration. An eighth note would have twice the duration of a sixteenth note, and a quarter note would have twice the duration of an eighth note, etc.

Figure 3.18 Sheet-music notation is a time–frequency diagram.

The vertical axis has a much more complicated notation to define frequency. If you look carefully at Fig. 3.18, you will see that the black dots that mark the notes lie either on one of the horizontal lines or in the space betweeen two lines. Each of these denotes a white key on the piano keyboard depicted in Fig 3.19. The black keys on the piano are denoted by "sharps" (♯) or flats" (♭). Figure 3.18 has a few notes sharped. The musical score is divided into a treble section (the top five lines) and a bass section (the bottom five lines). The vertical reference point for the notes is "middle C," which lies on an invisible horizontal line between the treble and bass sections (key number 40 in Fig. 3.19). Thus the bottom horizontal line in the treble section represents the white key (E) that is two above middle C; i.e. key number 44 in Fig. 3.19.

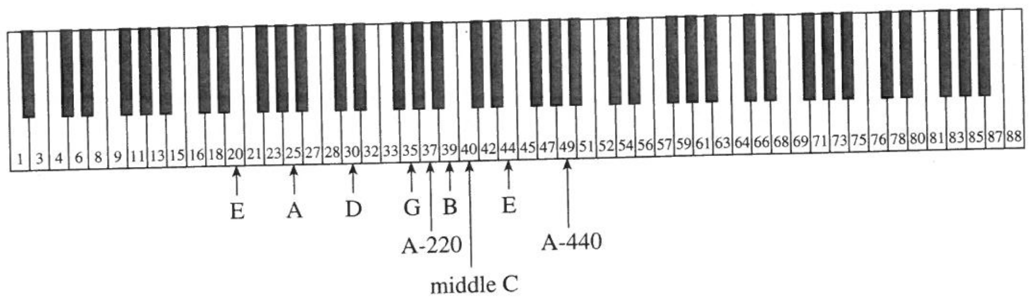

Figure 3.19 Piano keys can be numbered from 1 to 88. Middle C is key 40. A–440 is key 49.

Once the mapping from the musical score to the piano keys has been made, we can write a mathematical formula for the frequency. A piano, which has 88 keys, is divided into octaves containing twelve keys each. The meaning of the word *octave* is a doubling of the frequency. Within an octave, the neighboring keys maintain a constant frequency ratio. Since there are twelve keys per octave, the ratio (r) is

$$r^{12} = 2 \qquad \Longrightarrow \qquad r = 2^{1/12} = 1.0595$$

With this ratio, we can compute the frequencies of all keys if we have one reference. The convention is that the A key above middle C, called A–440, is at 440 Hz. Since A–440 is key number 49 and middle C is key number 40, the frequency of middle C is

$$f_{\text{middle C}} = 440 \times 2^{(40-49)/12} \approx 262 \text{ Hz}$$

CDROM

LAB: CHIRP SYNTHESIS

It is not our objective to explain how to read sheet music, although two of the lab projects will investigate methods for synthesizing waveforms to create songs and musical sounds. What is interesting about musical notation is that it uses a two-dimensional display to indicate frequency content that changes with time. If we adopt a similar notation, we can specify how to synthesize sinusoids with time-varying frequency content. Our notation is illustrated in Fig. 3.20.

3.5.1 Stepped Frequency

The simplest example of time-varying frequency content is to make a waveform whose frequency holds for a short duration and then steps to a higher (or lower) frequency. An example from music would be to play a "scale" that would be a succession of notes progressing over one octave. For example, the C-major scale consists of playing the notes {C, D, E, F, G, A, B, C} one after another, starting at middle C. This scale is played completely on the white keys. The frequencies of these notes are:

Middle C	D	E	F	G	A	B	C
262 Hz	294	330	349	392	440	494	523

Figure 3.20 should be interpreted as follows: Synthesize the frequency 262 Hz for 200 msec, then the frequency 294 Hz during the next 200 msec, and so on. The total waveform duration will be 1.6 sec. In music notation, the notes would be written as in Fig. 3.21(top), where each note is a quarter note.

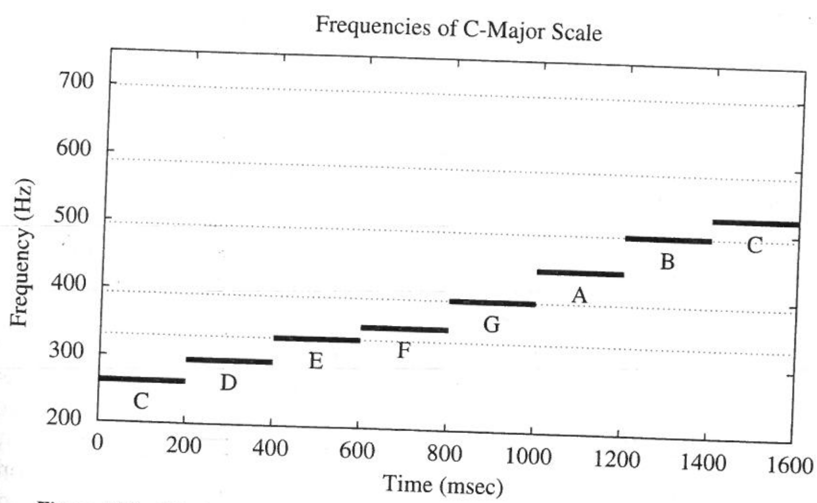

Figure 3.20 Ideal time-frequency diagram for playing the C-major scale. The horizontal dotted lines correspond to the five lines in the treble staff.

converters, respectively, and for purposes of this lab we will assume that they are perfect realizations.

1. The ideal C-to-D converter will be implemented in MATLAB by taking the formula for the continuous-time signal and evaluating it at the sample times, nT_s. This assumes perfect knowledge of the input signal, but for sinusoidal signals we have a mathematical equation for the continuous-time signal.

   To begin, compute a vector `x1` of samples of a sinusoidal signal with $A = 100$, $\omega_0 = 2\pi(1100)$, and $\phi = 0$. Use a sampling rate of 8000 samples/sec, and compute a total number of samples equivalent to 2 seconds' time duration. You may find it helpful to recall that the MATLAB statement `tt=(0:0.01:3);` would create a vector of numbers from 0 through 3 with increments of 0.01. Therefore, it is necessary only to determine the time increment needed to obtain 8000 samples in one second.[5]

   Using `sound( )`, play the resulting vector through the D-to-A converter of your computer, assuming that the hardware can support the $f_s = 8000$ Hz rate (or $f_s = 11{,}025$ Hz). Listen to the output.
2. Now compute a vector `x2` of samples (again with duration 2 secs) of the sinusoidal signal for the case $A = 100$, $\omega_0 = 2\pi(1650)$, and $\phi = \pi/3$. Listen to the signal reconstructed from these samples. How does it compare to the signal in item 1? Put both signals together in a new vector defined with the following MATLAB statement (assuming that both `x1` and `x2` are row vectors):

```
xx = [x1 zeros(1,2000) x2];
```

   Listen to this signal. Explain what you heard.
3. Now send the vector `xx` to the D-to-A converter again, but double the sampling rate in `sound( )` to 16,000 samples/sec. Do not recompute the samples in `xx`; just tell the D-to-A converter that the sampling rate is 16,000 samples/sec. Describe what you heard. Observe how the *duration* and *pitch* of the signal changed. Explain.

Instructor Verification (separate page)

C.3.2.3 Piano Keyboard Section C.3.3 of this lab will consist of synthesizing the notes of a well-known piece of music.[6] Since we will use sinusoidal tones to represent piano notes, a quick introduction to the frequency layout of the piano keyboard is

[5] Another popular rate is 11,025 samples/sec, which is one-fourth of the rate used in audio CD players.

[6] If you have little or no experience reading music, don't be intimidated. Only a little knowledge is needed to carry out this lab. On the other hand, the experience of working in an application area where you must quickly acquire knowledge is a valuable one. Many real-world engineering problems have this flavor, especially in signal processing, which has such a broad applicability in such diverse areas as geophysics, medicine, radar, speech, and the like.

needed (see Fig. C.2). On a piano, the keyboard is divided into octaves, the notes in each octave being twice the frequency of the notes in the next lower octave. For example, the reference note is the A above middle C, which is usually called A-440 (or A_5) because its frequency is 440 Hz. Each octave contains 12 notes (5 black keys and 7 white keys) and the ratio between the frequencies of the notes is constant between successive notes. Thus, this ratio must be $2^{1/12}$. Since middle C is 9 keys below A-440, its frequency is approximately 261.6 Hz. Consult Section 3.5 in Chapter 3 for more details.

MUSIC GUI

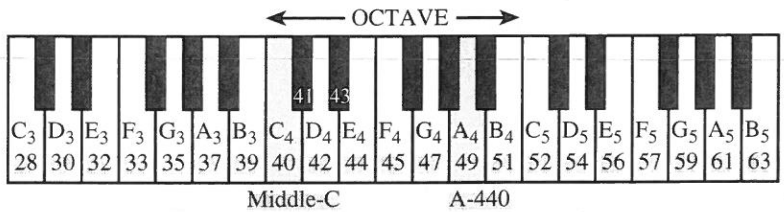

Figure C.2 Layout of a piano keyboard. Key numbers are shaded. The notation C_4 means the C key in the fourth octave.

Musical notation shows which notes are to be played, and their relative timing (half notes last twice as long as quarter notes, which, in turn, last twice as long as eighth notes). Figure C.3 shows how the keys on the piano correspond to notes drawn in musical notation.

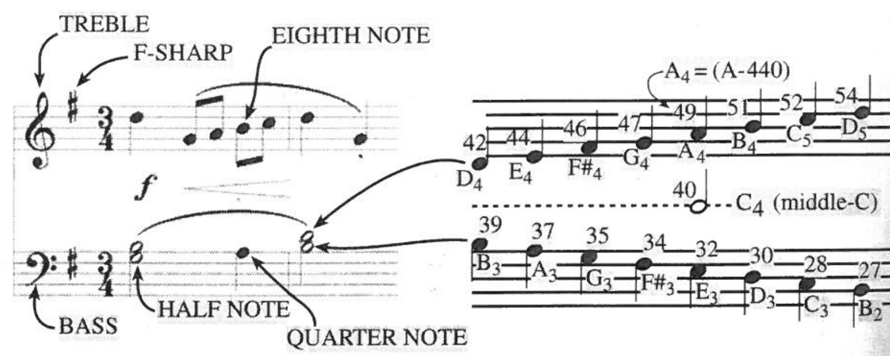

Figure C.3 Musical notation is a time-frequency diagram where vertical position indicates the frequency of the note to be played.

Another interesting relationship is the ratio of fifths and fourths as used in a chord. Strictly speaking, the fifth note should be 1.5 times the frequency of the base note. For middle C, the fifth is G, but the frequency of G is about 392 Hz, which

MUSIC GUI

is not exactly 1.5 times 261.6. It is very close, but the slight detuning introduced by the ratio $2^{1/12}$ gives a better sound to the piano overall. This innovation in tuning is called "equally tempered," and was introduced in Germany in the 1760s and made famous by J. S. Bach in *The Well-Tempered Clavichord*.

You can use the ratio $2^{1/12}$ to calculate the frequency of notes anywhere on the piano keyboard. For example, the E flat above middle C (key number 43) is 6 keys below A-440, so its frequency should be $f = 440 \times 2^{-6/12} = 440/\sqrt{2} \approx 311$ Hz.

1. Generate a sinusoid of 2 seconds' duration to represent the note E_5 above A-440 (key number 56). Choose the appropriate values for T_s and `fs`. Remember that `fs` should be at least twice as high as the frequency of the sinusoid you are generating. Also, T_s and `fs` must "match" in order for the note played out of the D-to-A converter to sound correct.

2. Now write an M-file to produce a desired note for a given duration. Your M-file should be in the form of a function called `note.m`. You may want to call the `sumcos` function that you wrote for Lab C.2. Your function should have the following form:

```
function tone = note(keynum,dur)
% NOTE  Produce a sinusoidal waveform corresponding to a
%       given piano key number
%
% usage:  tone = note (keynum, dur)
%
%        tone = the output sinusoidal waveform
%      keynum = the piano keyboard number of the desired note
%         dur = the duration (in seconds) of the output note
%
fs = 8000;  %-- use 11025 Hz on PC/Mac, 8000 on UNIX
tt = 0:(1/fs):dur;
freq =
tone =
```

For the `freq =` line, use the formulas based on $2^{1/12}$ to determine the frequency for a sinusoid in terms of its key number. You should start from a reference note (middle C or A-440 is recommended) and solve for the frequency based on this reference. For the `tone =` line, generate the actual sinusoid at the proper frequency and duration.

Figure C.8 The first few measures of the theme from *Beethoven's Fifth*.

Twinkle, Twinkle, Little Star

C.3.3.8 Alternative Piece: *Twinkle, Twinkle, Little Star* Follow the project description given in Section C.3.3.2, but use the piece *Twinkle, Twinkle, Little Star* written by Mozart. The first few measures are shown in Fig. C.9, and you can listen to the part that you will synthesize by following the links on the *DSP First* CD-ROM. More of the song can be found on the CD-ROM, where an entire page of the music is reproduced.

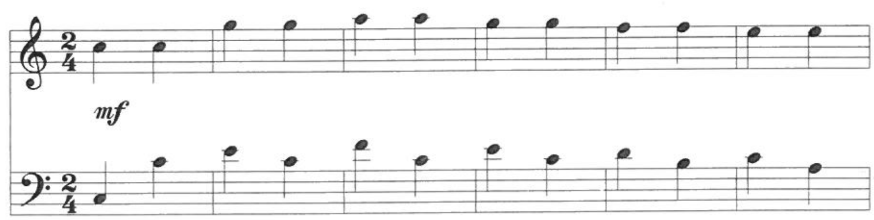

Figure C.9 The first few measures of *Twinkle, Twinkle, Little Star*.

C.3.4 Sound Evaluation Criteria

Here are some guidelines for evaluating the music synthesis projects

Does the file play notes? All Notes___ Most___ Treble only___

Overall Impression:

Excellent: Enjoyable sound, good use of extra features such as harmonics, envelopes, etc.

Good: Bass and treble clefs synthesized and in sync, few errors, one or two special features.

Average: Basic sinusoidal synthesis, including the bass, with only a few errors.

Poor: No bass notes, or treble and bass not synchronized, many wrong notes.