FM Synthesis for Musical Instruments: Bells and Clarinets

Pre-Lab: Read this handout before going to your assigned lab section with special attention to sections 1 and 2. Be prepared to define the key terms in equation 1 in your own words for the TA. The key terms are: modulating frequency, modulation index envelope, and instantaneous frequency. You will not be doing section 3 or 5 of this lab.

Verification: Show the TA your written definitions of the key terms at the beginning of the lab. The warm-up section and the bell sounds must be completed during your assigned lab time and demonstrated to your TA for verification.

Lab Report: Your report should include the results from Section 4.3, 4.4, and 6.5 with graphs and explanations. You need to label the axes of your plots and include a title for every plot. In order to keep track of plots, include your plot in-lined within your report. In your report, describe what you did and what your output sounded like. Include your Matlab code in an appendix.

1. Introduction

The objective of this lab is to introduce more complicated signals that are related to the basic sinusoid. These signals, which implement frequency modulation (FM) and amplitude modulation (AM), are widely used in communications systems such as radio and television. They can also be used to create interesting sounds that mimic musical instruments. The textbook CD-ROM provides examples of these signals for many different conditions.

2. Overview

Frequency modulation (FM) can be used to make interesting sounds which mimic musical instruments, such as bells, woodwinds, drums, etc. The goal of this lab is to implement some of these FM schemes and listen to the results. From the last lab, we know that FM defines the signal *x(t)* as a cosine with a time-varying angle

$x(t) = A\cos(\psi(t))$

And that the instantaneous frequency changes according to the time derivative of $\psi(t)$. If $\psi(t)$ is linearly increasing with time, *x(t)* is a constant-frequency sinusoid; whereas if $\psi(t)$ is quadratic in time, *x(t)* is a chirp signal whose frequency changes linearly in time. FM music synthesis uses a more interesting $\psi(t)$, one that is sinusoidal. Since the derivative of a sinusoid is also a sinusoid, the instantaneous frequency of *x(t)* will also oscillate. This is useful for synthesizing instrument sounds because the proper choice of the modulating frequencies will produce a fundamental frequency and several overtones.

The general equation for an FM sound synthesizer is:

$$x(t) = A(t)\cos\left(2\pi f_c t + I(t)\cos(2\pi f_m t + \phi_m) + \phi_c\right) \tag{1}$$

Where *A(t)* is the signal's time-varying amplitude. *A(t)* is a function of time so that the instrument sound can be made to fade out slowly or cut off quickly. Such a function is called an envelope. The constant parameter f_c is called the *carrier* frequency. Note that when you take the derivative of $\psi(t)$ to find the instantaneous frequency $f_i(t)$, the result is

$$\begin{aligned} f_i(t) &= \frac{1}{2\pi}\frac{d}{dt}\psi(t) \\ &= \frac{1}{2\pi}\frac{d}{dt}\left(2\pi f_c t + I(t)\cos(2\pi f_m t + \phi_m) + \phi_c\right) \\ &= f_c - I(t) f_m \sin(2\pi f_m t + \phi_m) + \frac{dI}{dt}\cos(2\pi f_m t + \phi_m) \qquad (2) \end{aligned}$$

Note that f_c is a constant in (2). It is the frequency that would be produced without any frequency modulation. The parameter f_m is called the *modulating* frequency. It expresses the rate of oscillation of $f_i(t)$. The parameters ϕ_m and ϕ_c are arbitrary phase constants, usually both set to $-\pi/2$ so that $x(0) = 0$.

The function $I(t)$ has a less obvious purpose than the other FM parameters in (1). It is technically called the *modulation index envelope.* To see what it does, examine the expression for the instantaneous frequency (2). The quantity $I(t)f_m$ multiplies a sinusoidal variation of the frequency. If $I(t)$ is constant or $\dfrac{dI}{dt}$ is relatively small, then $I(t)f_m$ gives the maximum amount by which the instantaneous frequency deviates from f_c. Beyond that, however, it is difficult to relate $I(t)$ to the sound made by $x(t)$ without some rather advanced mathematical analysis.

In our study of signals, we would like to characterize $x(t)$ as the sum of several constant-frequency sinusoids instead of a single signal whose frequency changes. In this regard, the following are facts that can be demonstrated experimentally: when $I(t)$ is small (e.g., $I \approx 1$), low multiples of the carrier frequency (f_c) have high amplitudes. When $I(t)$ is large $(I > 4)$, both low and high multiples of the carrier frequency have high amplitudes. The net result is that $I(t)$ can be used to vary the harmonic content of the instrument sound (called overtones). When $I(t)$ is small, mainly low frequencies will be produced. When $I(t)$ is large, higher harmonic frequencies can also be produced. Since $I(t)$ is a function of time, the harmonic content will change with time. For more details see the paper by Chowning.[1]

~~3 Warm-up~~

~~3.1 Chirps and Aliasing~~

~~Use your MATLAB function `mychirp` (from Lab 3) to synthesize a "chirp" signal with the following parameters:~~

1. ~~A total time duration of 2.5 secs. where the *desired* instantaneous frequency starts at 13,000 Hz and ends at 200 Hz.~~
2. ~~Use a sampling rate of $f_s = 8000$ Hz.~~

~~Listen to the signal. What comments can you make regarding the sound of the chirp (e.g., is it linear)? Does it chirp down, or chirp up, or both? Create a spectrogram of your chirp signal. Use the sampling theorem (from Chapter 4 in the text) to help explain what you hear and see.~~

~~In addition, make some theoretical calculations by hand: Determine the range of frequencies (in hertz) that will be synthesized by this MATLAB script. Make a sketch by hand of the instantaneous frequency versus time. Explain how *aliasing* affects the instantaneous frequency that is actually heard. Listen to the signal again to verify that it has the expected frequency content.~~

~~**Instructor Verification** (separate page)~~

[1]Ref: John M. Chowning, "The Synthesis of Complex Audio Spectra by means of Frequency Modulation," *Journal of the Audio Engineering Society*, vol. 21, no. 7, Sept. 1973, pp. 526–534.

~~3.2 Wideband FM~~

CD-ROM

Demos Spectrograms

~~Using a sampling rate of $f_s = 8000$ Hz, generate a set of wideband FM "chirps" with sinusoidal modulation according to the formula~~

$$x(t) = \cos(2\pi f_0 t + B\sin(2\pi f_m t))$$

~~evaluated over the time interval $0 \le t \le 1.35$ secs.~~

~~(a) First of all, do the case where $f_0 = 900$ Hz, $f_m = 3$ Hz and $B = 200$. Generate the signal, plot its spectrogram and listen to the signal to see if it corresponds to the spectrogram.~~

~~(b) Make a sketch (by hand) of the instantaneous frequency for the signal in part (a).~~
~~**Instructor Verification** (separate page)~~

~~(c) Now, change f_m to be $f_m = 30$ Hz and $B = 20$, but keep $f_0 = 900$ Hz. Generate the signal, plot its spectrogram and listen to the signal.~~

~~(d) One again, sketch (by hand) the instantaneous frequency for the signal in part (c) to see if it corresponds to the spectrogram.~~

~~(e) Now, change f_m to be $f_m = 300$ Hz and $B = 2$, but keep $f_0 = 900$ Hz. Generate the signal, plot its spectrogram and listen to the signal. This is a case of *wideband FM*, and the instantaneous frequency concept no longer explains the sound we hear, but you should write a reasonable explanation for the spectrum that you observe.~~
~~**Instructor Verification** (separate page)~~

When unsure about a command, use `help`.

4 FM Synthesis of Instrument Sounds

4.1 Generating the Bell Envelopes

Now we take the general FM synthesis formula (1) and specialize for the case of a bell sound. The amplitude envelope $A(t)$ and the modulation index envelope $I(t)$ for the bell are both decaying exponentials. That is, they both have the following form:

$$y(t) = e^{-t/\tau} \tag{3}$$

where τ is a parameter that controls the decay rate of the exponential. Notice that $y(0) = 1$ and $y(\tau) = 1/e$, so τ is the time it takes a signal of the form (3) to decay to $1/e = 36.8\%$ of its initial value. For this reason, the parameter τ is called the *time constant.*

Use (3) to write a MATLAB function that will generate a decaying exponential to be used later in synthesizing a bell sound. The file header should look like this:

```
function    yy = bellenv(tau, dur, fsamp);
%BELLENV    produces envelope function for bell sounds
%
%        usage: yy = bellenv(tau, dur, fsamp);
%
%               where tau = time constant
%                      dur = duration of the envelope
%                    fsamp = sampling frequency
%                  returns:
%                       yy = decaying exponential envelope
%
%   note: produces exponential decay for positive tau
```

The function will be one or two lines of MATLAB code. The first line should define your time vector based on `fsamp` and `dur`, and the second generates the exponential (3).

The bell's amplitude envelope, $A(t)$, and modulation index envelope, $I(t)$ are identical, up to a scale factor.

$$A(t) = A_0 e^{-t/\tau} \qquad \text{and} \qquad I(t) = I_0 e^{-t/\tau}$$

Hence, one call to the `bellenv` function will generate the shape for both envelopes.

4.2 Parameters for the Bell

Now that we have the bell's amplitude and modulation index envelopes, we can create the actual sound signal for the bell by specifying all the parameters in the general FM synthesis formula (1). The frequencies f_c and f_m must be given numerical values. The ratio of carrier to modulating frequency is important in creating the sound of a specific instrument. For the bell, a good choice for this ratio is 1:2, e.g., $f_c = 110$ Hz and $f_m = 220$ Hz.

Now write a simple M-file `bell.m` that implements (1) to synthesize a bell sound. Your function should call `bellenv.m` to generate $A(t) = A_0 e^{-t/\tau}$ and $I(t) = I_0 e^{-t/\tau}$.

```
function  xx = bell(ff, Io, tau, dur, fsamp)
%BELL     produce a bell sound
%
%     usage:   xx = bell(ff, Io, tau, dur, fsamp)
%
%     where:  ff = frequency vector (containing fc and fm)
%             Io = scale factor for modulation index
%            tau = decay parameter for A(t) and I(t)
%            dur = duration (in sec.) of the output signal
%          fsamp = sampling rate
```

4.3 The Bell Sound

Test your `bell( )` function using the parameters of case #1 in the table. Play it with the `soundsc()` function at 11,025 Hz.[2] Does it sound like a bell? The value of $I_0 = 10$ for scaling the modulation index envelope is known to give a distinctive sound. Later on, you can experiment with other values to get a variety of bells.

[2] A higher sampling rate of 11,025 Hz is used because the signal contains many harmonics, some of which might alias if a lower f_s were used. You should experiment with lower values of f_s to see if you can hear a difference, e.g., $f_s = 8000$ Hz.

CASE	f_c (Hz)	f_m (Hz)	I_0	τ (sec)	T_{dur} (sec)	f_s (Hz)
1	110	220	10	2	6	11,025
2	220	440	5	2	6	11,025
3	110	220	10	12	3	11,025
4	110	220	10	0.3	3	11,025
5	250	350	5	2	5	11,025
6	250	350	3	1	5	11,025

The frequency spectrum of the bell sound is very complicated, but it does consist of spectral lines, which can be seen with a spectrogram. Among these frequencies, one spectral line will dominate what we hear. We would call this the *note frequency* of the bell. It is tempting to guess that the note frequency will be equal to f_c, but you will have to experiment to find the true answer. It might be f_m, or it might be something else—perhaps the fundamental frequency which is the greatest common divisor of f_c and f_m.

For each case in the table, do the following:

(a) Listen to the sound by playing it with the `soundsc()` function.

(b) Calculate the fundamental frequency of the “note” being played. Explain how you can verify by listening that you have the correct fundamental frequency.

(c) Describe how you can hear the frequency content changing according to $I(t)$. Plot $f_i(t)$ versus t for comparison.

(d) Display a spectrogram of the signal. Describe how the frequency content changes, and how that change is related to $I(t)$. Point out the “harmonic” structure of the spectrogram, and calculate the fundamental frequency, f_0.

(e) Plot the entire signal and compare it to the envelope $A(t)$ generated by `bellenv`.

(f) Plot about 100–200 samples from the middle of the signal and explain what you see, especially the frequency variation.

If you are making a lab report, do the plots for two cases—choose one of the first four and one of the last two. Write up an explanation only for the two that you choose.

4.4 Comments about the Bell

Cases #3 and #4 are extremes for choosing the decay rate τ. In case #3, the waveform does not decay very much over the course of three seconds and sounds a little like a sum of harmonically related sinusoids. With a “faster” decay rate, as in case #4, we get a percussion-like sound. Modifying the fundamental frequency f_0 (determined in part (d) above) should have a noticeable effect on the tone you hear. Try some different values for f_0 by changing f_c and f_m, but still in the ratio of 1:2. Describe what you hear.

Finally, experiment with different carrier to modulation frequency ratios. For example, in his paper, Chowning uses a fundamental frequency of $f_0 = 40$ Hz and a carrier to modulation frequency ratio of 5:7. Try this and a few other values. Which parameters sound “best” to you?

~~5 Optional: C-Major Scale~~

~~Finally, synthesize other note frequencies. For example, try to make the C-major scale (defined in Lab 3) consisting of seven consecutive notes.~~

6 Woodwinds

As an alternative to the bell sounds, this section shows how different parameters in the same FM synthesis formula (1) will yield a clarinet sound, or other woodwinds.

6.1 Generating the Envelopes for Woodwinds

CD-ROM
woodwenv.m

There is a function on the CD-ROM called `woodwenv` which produces the functions needed to create both the $A(t)$ and $I(t)$ envelopes for a clarinet sound. The file header looks like this:

```
function     [y1, y2] = woodwenv(att, sus, rel, fsamp)
%WOODWENV        produce normalized amplitude and modulation index
%                functions for woodwinds
%
%       usage: [y1, y2] = woodwenv(att, sus, rel, fsamp);
%
%        where  att = attack TIME
%               sus = sustain TIME
%               rel = release TIME
%             fsamp = sampling frequency (Hz)
%        returns:
%             y1 = (NORMALIZED) amplitude envelope
%             y2 = (NORMALIZED) modulation index envelope
%
%  NOTE: attack is exponential, sustain is constant,
%          release is exponential
```

The outputs from `woodwenv` are normalized so that the minimum value is zero and the max is one. Try the following statements to see what the function produces:

```
fsamp = 8000;
Ts = 1/fsamp;
tt = delta : Ts : 0.5;
[y1, y2] = woodwenv(0.1, 0.35, 0.05, fsamp);
subplot(2,1,1), plot(tt,y1), grid on
subplot(2,1,2), plot(tt,y2), grid on
```

6.2 Scaling the Clarinet Envelopes

Since the woodwind envelopes produced by `woodwenv` range from 0 to 1, some scaling is necessary to make them useful in the FM synthesis equation (1). In this section, we consider the general process of linear re-scaling. If we start with a *normalized* signal $y_{\text{norm}}(t)$ and want to produce a new signal whose max is $y_{\max}$ and whose min is $y_{\min}$, then we must map 1 to $y_{\max}$ and 0 to $y_{\min}$. Consider the linear mapping:

$$y_{\text{new}}(t) = \alpha\, y_{\text{norm}}(t) + \beta \tag{4}$$

Determine the relationship between α and β and $y_{\max}$ and $y_{\min}$, so that the max and the min of $y_{\text{new}}(t)$ are correct.
Test this idea in MATLAB by doing the following example (where $\alpha = 5$ and $\beta = 3$):

```
ynorm = 0.5 + 0.5*sin( pi*[0:0.01:1]);
subplot(2,1,1),  plot(ynorm)
alpha = 5;    beta = 3;
ynew = alpha*ynorm + beta;          %<------ Linear re-scaling
subplot(2,1,1),  plot(ynew)
max(ynorm),  min(ynorm)      %<--- ECHO the values
max(ynew),  min(ynew)
```

What happens if we make α negative?
Write a short one-line function that implements (4) above. Your function should have the following form:
`function y = scale(data, alpha, beta).`

6.3 Clarinet Envelopes

For the clarinet sound, the amplitude $A(t)$ needs no scaling—the MATLAB function `sound` will automatically scale to the maximum range of the D/A converter. Thus, $A(t)$ equals the vector `y1`. From the plot of `y1` shown in Fig. 1, it should be obvious that this envelope will cause the sound to rise quickly to a certain volume, sustain that volume, and then quickly turn off.

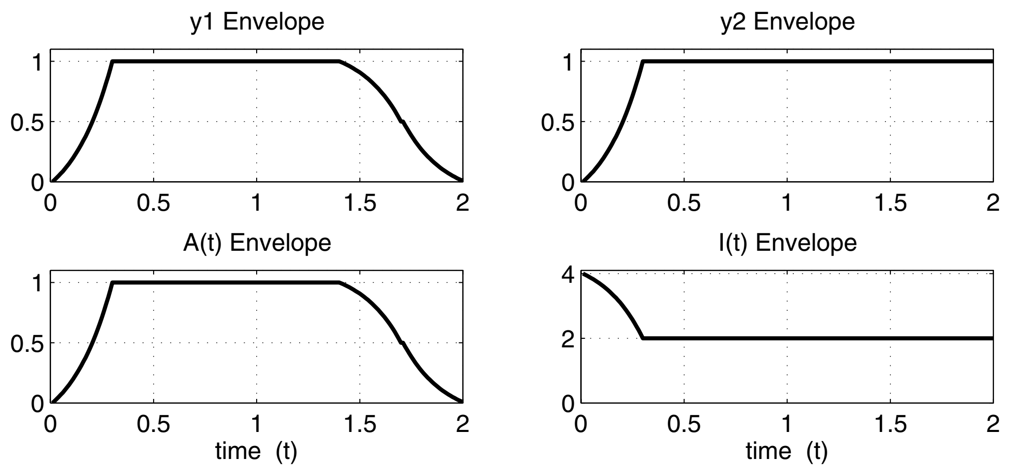

Figure 1: Envelopes for the woodwinds. The functions $A(t)$ and $I(t)$ are produced by scaling `y1` and `y2`, the outputs of `woodwenv.`

The modulation index envelope, $I(t)$, however, does not equal `y2`. The range for $I(t)$ lies between 2 and 4 as in Fig. 1. Furthermore, there is an inversion so that when `y2` is zero, $I(t)$ should equal 4, and when `y2` is one, $I(t)$ should be 2. Using this information solve for the appropriate α and β then use `scale` to produce the modulation index envelope function (`I`) for a clarinet sound.

6.4 Parameters for the Clarinet

So far we have a general equation for FM signals, an amplitude envelope for the clarinet, and a modulation index envelope for the clarinet. To create the actual sound signal for the clarinet, we need to specify the additional parameters in (1). The ratio of carrier to modulating frequency is important in creating the sound of a specific instrument. For the clarinet, this ratio should be 2:3. The actual note frequency will be the greatest common divisor of the carrier and modulating frequencies. For example, when we choose $f_c = 600$ Hz and $f_m = 900$ Hz, the synthesized signal will have a fundamental frequency of $f_0 = 300$ Hz.

Write a simple M-file `clarinet.m` that implements the FM synthesis equation (1) to synthesize a clarinet note. Your function should generate the envelopes $A(t)$ and $I(t)$ using textttscale and texttttwoodwenv. The function header should look like this:

```
function  yy = clarinet(f0, Aenv, Ienv, dur, fsamp)
%CLARINET     produce a clarinet note signal
%
%     usage:  yy = clarinet(f0, Aenv, Ienv, dur, fsamp)
%
%     where:   f0 = note frequency
%            Aenv = the array holding the A(t) envelope
%            Ienv = the array holding the I(t) envelope
%             dur = the amount of time the signal lasts
%           fsamp = the sampling rate
```

6.5 Experiment with the Clarinet Sound

Using your `clarinet( )` function, create a 250 Hz clarinet note with $f_s = 8000$ or 11,025 Hz. Play it with the `sound( xnote, fs )` function. Does it sound like a clarinet? How can you verify that its fundamental frequency is at 250 Hz?

Explain how the modulation index $I(t)$ will affect the frequency content versus time of the clarinet sound. Describe how you can hear the frequency content changing according to $I(t)$? Plot the instantaneous frequency $f_i(t)$ versus t for comparison.

Plot the entire signal and compare it to the amplitude envelope function `y1` in Fig. 1. Plot about 100–200 samples from the middle of the signal and explain what you see.

FM Synthesis Lab Validation Page

Name:
Lab Session:

Pre-lab:
Definitions of key terms in student's own words (modulating frequency, modulation index envelope, and instantaneous frequency) showed to TA:____________________

Bell sounds:
Played two different bell sounds using matlab routine *bell*:_________________________
Demonstrate expertise using functions with parameters:___________________________

Clarinet or other woodwinds:
Demonstrate function *clarinet*:____________________________