EE 313 Linear Signals & Systems (Fall 2018)

Solution Set for Mini-Project #1 on FM Synthesis for Musical Instruments

Mr. Houshang Salimian & Prof. Brian L. Evans

1.0 Introduction (*6 points*)

Amplitude modulation (AM) and frequency modulation (FM) are commonly used in transmitting signals in radio frequencies, such AM radio and Bluetooth networks. Another interesting application of these modulation methods is to synthesize complex audio signals including notes that sound as if they had been played by a musical (acoustic) instrument. When using a sinusoid signal with a constant frequency in the audible frequency range, we can make an artificial sounding note, but it is missing a rich set of harmonics that are characteristic of an acoustic instrument. By adding frequencies to the note frequency, we might hear a more complicated sound signal, but it is not necessarily mimicking the sound of a musical instrument.

Another approach to make a signal that is more complicated than a sinusoidal signal at a fixed frequency is to use a chirp signal. In a chirp signal, the instantaneous frequency varies with time linearly so we will hear a sound whose principal frequency will gradually increase or decrease. Using frequency modulation, we can create signals that not only have a principal frequency varying with time but also have a rich harmonic structure. In this mini-project we will use the frequency modulation to make complex audio signals such as a bell sound. Also, we will check effect of different factors in frequency modulation on the audio signal that we synthesize.

2.0 Overview (*13 points*)

As mentioned in the Introduction, FM is a tool for creating interesting sounds. In this modulation, the angle of sinusoid varies nonlinearly with time, and instantaneous frequency covers the preferred range of frequency domain. Following is the equation that defines FM.

$$y(t) = A(t) \cos(2 \pi f_c t + I(t) \cos(2 \pi f_m t + \varphi_m) + \varphi_c)$$

In this project we will use f_c, f_m, $I(t)$, and $A(t)$ to observe their effects on a synthesized audio signal. Here, f_c is the carrier frequency, which is a constant value and can be detected when we hear the sound. Also, f_m is modulating frequency and determines rate of oscillations around f_c. I(t) is modulation index envelope and we can see its effect in the instantaneous frequency. Finally, A(t) is the time-varying amplitude, which we will use to model how the amplitude of a note played by a musician would vary over time.

The instantaneous frequency will be of particular interest in this project. It is defined as the derivative of the angle of the cosine function with respect to time divided by 2 π:

$$f_i(t) = \frac{1}{2\pi}\frac{d}{dt}(2\,\pi\, f_c\, t \,+\, I(t)\cos(2\,\pi\, f_m\, t + \,\varphi_m) + \,\varphi_c)$$

$$f_i(t) = f_c - I(t)\, f_m \sin(2\,\pi\, f_m\, t + \,\varphi_m) + \frac{d}{dt}I(t)(\cos(2\,\pi\, f_m\, t + \,\varphi_m))$$

Here, f_c is the constant value that the instantaneous frequency oscillates around. By increasing f_m, rate of oscillation will increase and we can detect a sound with sharper variations. If we select $I(t) = 0$, the generated signal will be an audio signal with a single principal frequency, while by selecting a non-zero constant for $I(t)$, the last term in $f_i(t)$ disappears. When $I(t)$ is a constant or when $I'(t)$ is relative small, the product of $I(t)$ and f_m determines the minimum and maximum instantaneous frequency.

Using frequency modulation, we can have instantaneous frequencies that are not equal to f_c or f_m, and the sound generated will contain many frequencies in its spectra. For example, by selecting f_m large enough we can make a wideband of instantaneous frequencies to synthesize a bell sound.

3.0 Warm-Up (*36 points*)

The previous sections discussed the use of frequency modulation to synthesize sound played by certain acoustic instruments. Frequency modulation changes the frequency content according to a message (input) signal $x(t)$ to produce the output $y(t) = \cos(2\,\pi\, f_c\, t + x(t))$.

The first case was that of a chirp signal that increases its principal instantaneous frequency linearly with time. The message (input) signal is a quadratic function of time $\pi\, f_{step}\, t^2$, which is further explored in Section 3.1. In the second case, the message (input) signal is a sinusoid of the form $B \sin(2\,\pi\, f_m\, t)$, which is further explored in Section 3.2.

3.1 Chirps and Aliasing

A chirp signal has the form

$$y(t) = \cos(\,\theta(t)\,) \text{ where } \theta(t) = 2\,\pi\,(\,f_c + \tfrac{1}{2} f_{step}\, t\,)\, t = 2\,\pi\, f_c\, t + \pi\, f_{step}\, t^2$$

The principal frequency is f_c when $t = 0$ and then changes over time at a rate of f_{step} in units of Hz/s. The principal frequency of a sinusoid at a given point in time is called the *instantaneous frequency*, and it is defined as $d\theta(t) / dt$ in units of rad/s. Here, $d\theta(t) / dt = 2\,\pi\, f_c + 2\,\pi\, f_{step}\, t = 2\,\pi\,(f_0 + f_{step}\, t)$. We can view the chirp signal as the output of a system $y(t) = \cos(2\,\pi\, f_c\, t + x(t))$ where the input is $x(t) = \pi\, f_{step}\, t^2$.

For this section, we'll be analyzing a chirp signal that

1. Has a total time duration of 2.5s where the desired instantaneous frequency starts at 13,000 Hz and ends at 200 Hz.
2. Is sampled at a sampling rate of $f_s = 8000$ Hz.

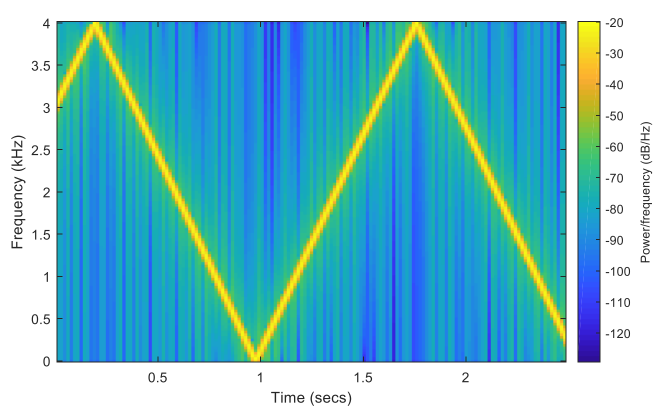
This is a downsweeping chirp where

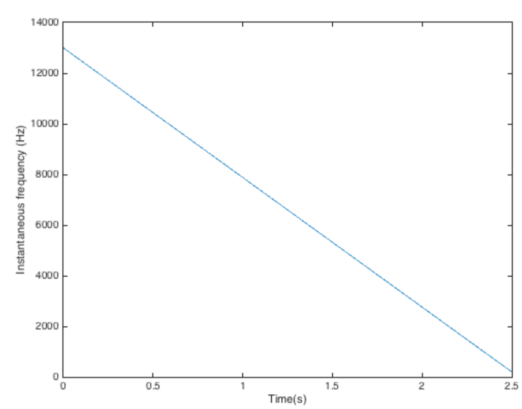
$$f_{step} = \frac{f_{end} - f_{start}}{T_{duration}} = \frac{200\text{ Hz} - 13000\text{ Hz}}{2.5s} = -5120\text{ Hz}^2$$

Converting the above equations into Matlab code:

```
fs = 8000;
Ts = 1/fs;
t_max = 2.5;
t = 0: Ts: t_max;

f_start = 13000;
f_end = 200;
f_step = (f_end - f_start) / t_max;
x = pi*f_step*(t.^2);
y = cos(2*pi*f_start*t + x);
f_instant = f_start + f_step*t;   %instantaneous frequency

sound(y, fs)

figure(1);
spectrogram(y, hamming(256), 128, 256, fs, 'yaxis');

figure(2);
plot(t, f_instant);
xlabel('Time(s)');
ylabel('Instantaneous frequency (Hz)');
xlim([0 2.5]);
```

According the equation for the chirp signal in continuous time, one should hear a sound with linearly decreasing frequency from 13000 Hz to 200 Hz. When listening to the signal, however, the principal frequency increases, then decreases, then increases, and finally decreases. From the spectrogram, the principal frequency initially increases from 3000 Hz to 4000 Hz, then falls to zero, then rises to 4000 Hz and finally falls to 200 Hz.





One is hearing via audio playback and seeing via the spectrogram plot the effects of sampling. The Sampling Theorem states that if a continuous-time signal is sampled at a rate $f_s > 2\ f_{max}$, where f_{max} is the maximum frequency in the continuous-time signal, then it is possible to reconstruct the continuous-time signal from its samples. Dividing both sides of the inequality by two yields $f_{max} < ½ f_s$. When sampling, the maximum frequency of the continuous-time signal that can be captured is up to but not necessarily including $½ f_s$. When including negative frequencies, frequencies in the interval $[-½ f_s, ½ f_s]$ are captured when sampling.

Let's first take a look at what happens at continuous time frequency f_0 with a sampling rate of f_s:

$$x[n] = \cos(2\pi f_0 t)|_{t=nT_s} = \cos\big(2\pi f_0 (nT_s)\big) = \cos\left(2\pi \frac{f_0}{f_s} n\right)$$

The discrete-time frequency is

$$\hat{\omega} = 2\pi \frac{f_0}{f_s}$$

The continuous-time frequency of $-½ f_s$ becomes a discrete-time frequency of $-\pi$, and a continuous-time frequency of $½ f_s$ becomes a discrete-time frequency of π. This means that through sampling, the range of discrete-time frequencies is on the interval $[-\pi, \pi]$.

Using f_0 = 13000 Hz and f_s = 8000 Hz,

$$x[n] = \cos\left(2\pi \frac{13000 \text{ Hz}}{8000 \text{ Hz}} n\right) = \cos\left(2\pi \frac{13}{8} n\right) = \cos\left(2\pi \frac{16}{8} n - 2\pi \frac{3}{8} n\right)$$

$$x[n] = \cos\left(4\pi n - 2\pi \frac{3}{8} n\right) = \cos\left(-2\pi \frac{3}{8} n\right) = \cos\left(2\pi \frac{3}{8} n\right)$$

Discrete-time frequency $2\pi \frac{3}{8}$ with f_s = 8000 Hz means a continuous-time frequency of 3000 Hz.

At the beginning of the continuous-time chirp signal when the principal continuous-time frequencies are decreasing from 13000 Hz to 12000 Hz, they actually appear to be increasing from 3000 Hz to 4000 Hz due to sampling at 8000 Hz. Aliasing continues to occur until the principal continuous-time frequencies in the chirp signal fall below 4000 Hz, which is $½ f_s$.

3.2 Wideband FM

This section analyzes the wideband frequency modulated (FM) signals to be used in Section 4 to synthesize notes being played by certain acoustic instruments. The signals have the form

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

This is equivalent to sending an input signal $x(t) = B \sin(2\pi f_m t)$ into a system with an output of $y(t) = \cos(2\pi f_0 t + x(t))$.

By taking derivative of the angle of the cosine term in $y(t)$ with respect to time, we can compute the instantaneous frequency

$$f_i(t) = \frac{1}{2\pi} \frac{d}{dt} (2\pi f_0 t + B \sin(2\pi f_m t)) = f_0 + B f_m \cos(2\pi f_m t)$$

In parts (a)-(f) below, we will plot the spectrogram and instantaneous frequency of $y(t)$.

In parts (a)-(f), we will also use the spectrogram to visualize the harmonic structure of $y(t)$.

We can unlock the harmonic structure in $y(t) = \cos(2\pi f_0 t + B\sin(2\pi f_m t))$ by first using a trigonometric identity, then using a Taylor series for $\cos(\theta)$ and $\sin(\theta)$ and finally determining the resulting harmonics. We give those three steps next.

Step #1: We'll use the following trigonometric identity

$\cos(\alpha+\beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$

to give

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

The term $\cos(2\pi f_0 t)\cos(B\sin(2\pi f_m t))$ is (sinusoidal) amplitude modulation that was analyzed in homework problem 2.2. Multiplication by $\cos(2\pi f_0 t)$ will cause a shift in the frequency components of $\cos(B\sin(2\pi f_m t))$ by $+f_0$ and $-f_0$.

We can perform a similar analysis of the term $\sin(2\pi f_0 t)\sin(B\sin(2\pi f_m t))$. This term is also a form of (sinusoidal) amplitude modulation that uses $\sin(2\pi f_0 t)$ instead of $\cos(2\pi f_0 t)$ that was analyzed in homework problem 2.2. A similar effect happens. Multiplying by $\sin(2\pi f_0 t)$ will cause a shift in the frequency components of $\sin(B\sin(2\pi f_m t))$ by $+f_0$ and $-f_0$.

Step #2: The frequency components in $\cos(B\sin(2\pi f_m t))$ were explored in homework problem 2.4(d). The solution for homework problem 2.4(d) used the Taylor series expansion of $\cos(\theta)$ which is

$$\cos(\theta) = 1 - \frac{1}{2!}\theta^2 + \frac{1}{4!}\theta^4 - \cdots$$

to obtain

$$\cos(B\sin(2\pi f_m t)) = 1 - \frac{1}{2!}B^2\sin^2(2\pi f_m t) + \frac{1}{4!}B^4\sin^4(2\pi f_m t) - \cdots$$

where $\sin^2(2\pi f_m t) = \frac{1}{2} - \frac{1}{2}\cos(2\pi(2f_m)t)$ which has frequency components of $-2f_m$, 0, and $2f_m$. We can view $\sin^4(2\pi f_m t) = \sin^2(2\pi f_m t)\sin^2(2\pi f_m t)$ which gives frequency components of $-4f_m$, $-2f_m$, 0, $2f_m$, and $4f_m$. If we include all of the higher-order terms, then we'll get all of the even harmonics of f_m.

In a similar way, we can expand $\sin(B\sin(2\pi f_m t))$ using the Taylor series

$$\sin(\theta) = \theta - \frac{1}{3!}\theta^3 + \frac{1}{5!}\theta^5 - \cdots$$

to obtain

$$\sin(B\sin(2\pi f_m t)) = B\sin(2\pi f_m t) - \frac{1}{3!}B^3\sin^3(2\pi f_m t) + \frac{1}{5!}B^5\sin^5(2\pi f_m t) - \cdots$$

If we keep all of the terms in the series, then we'll get all of the odd harmonics of f_m.

Step #3: By combining the results in steps 1 and 2,

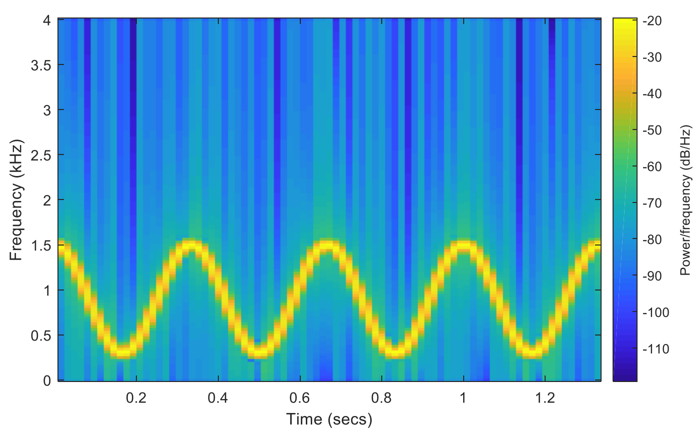
$y(t) = \cos(2\pi f_0 t + B\sin(2\pi f_m t)) = \cos(2\pi f_0 t)\cos(B\sin(2\pi f_m t)) - \sin(2\pi f_0 t)\sin(B\sin(2\pi f_m t))$

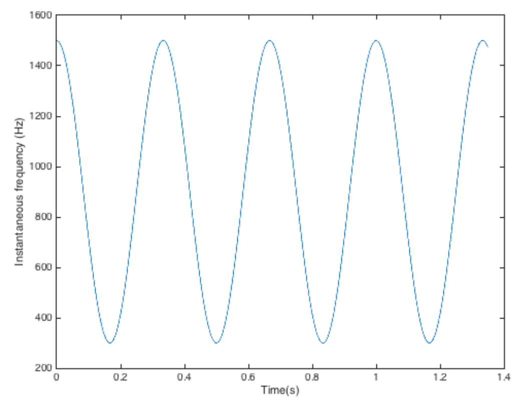
has frequency components of ..., $f_0 - 2f_m$, $f_0 - f_m$, f_0, $f_0 + f_m$, $f_0 + 2f_m$, ... That's what we see in the spectrograms below.

Section 3.2(a) and (b): Message frequency f_m = 3 Hz and modulus B = 200

Following is the Matlab code for this part.

```
fs = 8000;
Ts = 1/fs;
t_max = 1.35;
t = 0 : Ts : t_max;
f0 = 900;

fm = 3;
B = 200;

y = cos(2*pi*f0*t + B*sin(2*pi*fm*t));
sound(y, fs)

f_instant = f0 + B*fm*cos(2*pi*fm*t);

figure(1);
spectrogram(y, hamming(256), 128, 256, fs, 'yaxis');

figure(2);
plot(t, f_instant);
xlabel('Time(s)');
ylabel('Instantaneous frequency (Hz)');
```





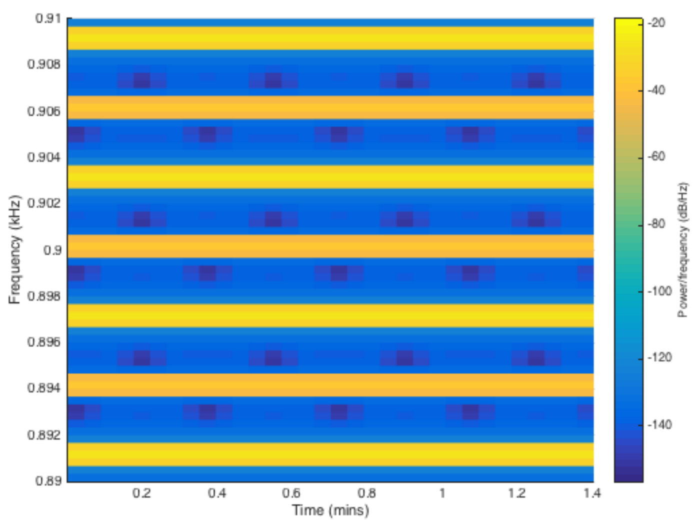
A block size of 256 is used for the spectrogram. A block size of 128 and 512 also work. For block sizes larger than 512, I would need to increase the signal duration by the same factor; otherwise, the samples in the signal would only allow a few segments (blocks) of samples to be computed for the spectrogram. The sound matches with spectrogram. It oscillates around 900 Hz and its frequency goes up to 1500 Hz and decreases to 300 Hz.

We can also use the spectrogram to visualize the harmonics present in the signal. The signal duration of 1.35s corresponds to 10800 samples, i.e. 8000 samples/s times 1.35s. From the previous mathematical analysis, the harmonics in the signal are spaced apart by f_m = 3 Hz. To obtain an accuracy of 1 Hz in the spectrogram, one would need to use a block size equal to the number of samples per second, which is 8000 in this case. One would need to generate a much longer signal to have enough segments (blocks) of samples to see the harmonics.

```
% Mini-Project #1
% Problem 3.2(a)
```

```
% N = block size for spectrogram.
% Frequency resolution is fs / N.
% Using a larger block size means
% that we'll need more samples for
% the signal (i.e., increase tmax).
% Ninc the factor to increase both
% tmax and N for the analysis.
fs = 8000;
Ts = 1/fs;
Ninc = 64;
tmax = 1.35*Ninc;
t = 0 : Ts : tmax;
f0 = 900;
fm = 3;
B = 200;
y = cos(2*pi*f0*t + B*sin(2*pi*fm*t));
N = 3*fs;
spectrogram(y, hamming(N), N/8, N, fs, 'yaxis');
ylim( [0.890 0.910] );
```

Here is the spectrogram zoomed into the interval of frequencies between 890 Hz and 910 Hz which shows the spacing between harmonics of $f_m = 3$ Hz:



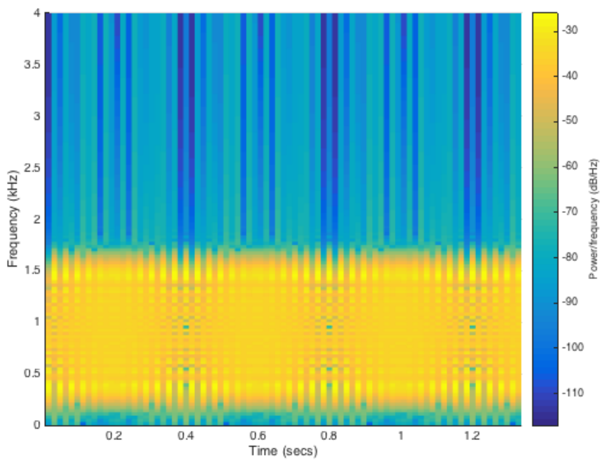
Section 3.2(c) and (d): Message frequency $f_m = 30$ Hz and modulus $B = 20$

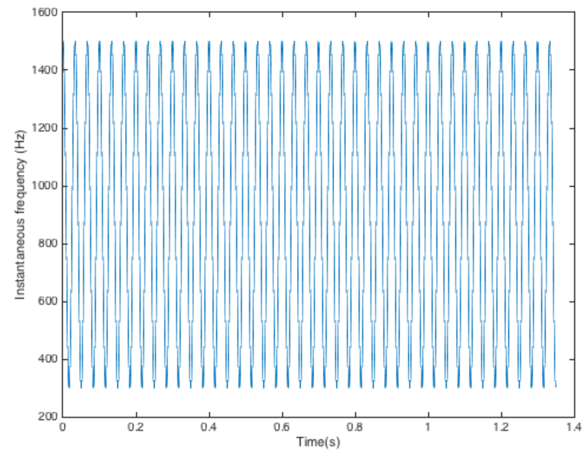
We repeat the above analysis in Sections 3.2(a) and (b) with $f_m = 30$ Hz and $B = 20$.

```
fs = 8000;
Ts = 1/fs;
t_max = 1.35;
t = 0 : Ts : t_max;
f0 = 900;

fm = 30;
B = 20;

y = cos(2*pi*f0*t + B*sin(2*pi*fm*t));
```

```
sound(y, fs)

f_instant = f0 + B*fm*cos(2*pi*fm*t);

figure(1);
spectrogram(y, hamming(256), 128, 256, fs, 'yaxis');

figure(2);
plot(t, f_instant);
xlabel('Time(s)');
ylabel('Instantaneous frequency (Hz)');
```

The instantaneous frequency oscillates between 300 Hz and 1500 Hz. When played as sound, the principal frequency changes faster that in parts (a) and (b) due to f_{m} of 30 Hz instead of 3 Hz.

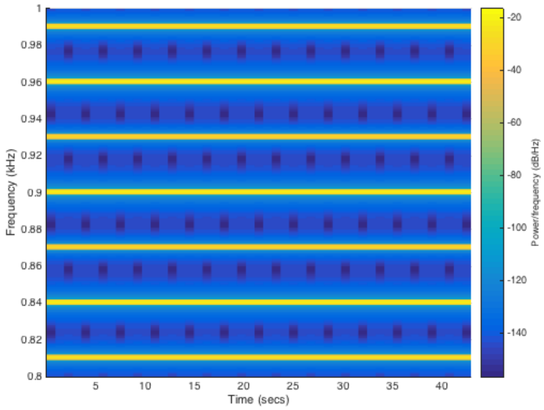




We can visualize the harmonics by generating a longer signal in the time domain:

```
% Mini-Project #1
% Problem 3.2(c)
% N = block size for spectrogram.
% Frequency resolution is fs / N.
% Using a larger block size means
% that we'll need more samples for
% the signal (i.e., increase tmax).
% Ninc the factor to increase both
% tmax and N for the analysis.
fs = 8000;
Ts = 1/fs;
Ninc = 32;
tmax = 1.35*Ninc;
t = 0 : Ts : tmax;
f0 = 900;
fm = 30;
B = 20;
y = cos(2*pi*f0*t + B*sin(2*pi*fm*t));
N = fs;
spectrogram(y, hamming(N), N/8, N, fs, 'yaxis');
ylim( [0.8 1.0] );
```

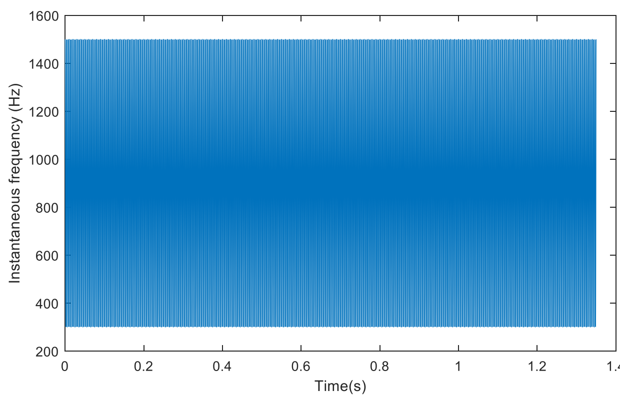
Zooming into 800-1000 Hz, the spectrogram reveals a harmonic spacing of f_m = 30 Hz:

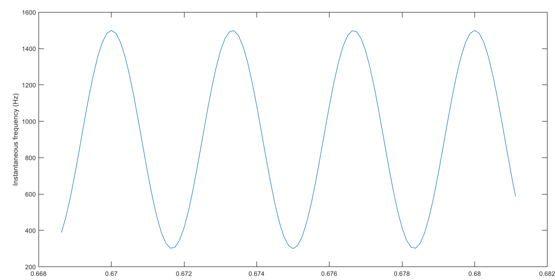


Section 3.2(e) and (f): Message frequency f_m = 300 Hz and modulus B = 2

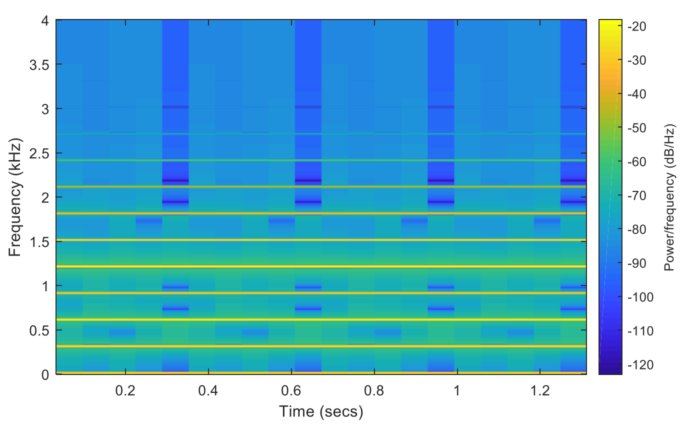
We repeat the above analysis in Sections 3.2(a) and (b) with f_m = 300 Hz and B = 2.

Instantaneous frequency is plotted below over a duration of 1.35s (right) and zoomed in (left):





We see harmonics shown by bright yellow lines at f_0 = 900 Hz, $f_0 + f_m$ = 1200 Hz, f_0 - f_m = 600 Hz, etc. That is, we see harmonics of 300 Hz plus an offset of 900 Hz up to ½ f_s :



4.0 FM Synthesis of Instrument Sounds (*27 points*)

4.1 Generating the Bell Envelopes

We will now customize the general FM synthesis equation to synthesize a bell sound:

$$y(t) = A(t)\cos(2\pi f_c t + I(t)\cos(2\pi f_m t + \varphi_m) + \varphi_c)$$

When a hammer strikes a bell, the bell will have a strong response in amplitude (volume) which will then decay will time. Accordingly, the lab project models the time-varying amplitude as a decaying exponential, and in addition, also models $I(t)$ in a similar way

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

where τ is the time constant. The same time constant is used in both models.

Function should be written as follows in Matlab.

```
function yy = bellenv(tau, dur, fsamp)
Tsamp = 1/fsamp;
t = 0:Tsamp:dur;
yy = exp(-t./tau);
end
```

This function is saved as bellenv.m and can be called by other m-files or in the command window. By inserting the input values, the envelope will be made to be used as $A(t)$ or $I(t)$.

4.2 Parameters for the Bell

Function should be written as follows in Matlab.

```
function x = bell(ff, Io, tau, dur, fsamp)
Tsamp = 1/fsamp;
t = 0:Tsamp:dur;
e1 = bellenv(tau, dur, fsamp);
x = e1.*cos(2*pi*ff(1)*t+Io*e1.*cos(2*pi*ff(2)*t));
end
```

This function is saved as bell.m and can be called in other m-files or in the command window.

4.3 The Bell sound

Here are several possible sets of parameters to use FM synthesis to generate a bell sound:

Case	f_c (Hz)	f_m (Hz)	I_0	τ (sec)	T_{dur} (sec)	f_s (Hz)
1	110	220	10	2	6	11025
2	220	440	5	2	6	11025
3	110	220	10	12	3	11025
4	110	220	10	0.3	3	11025
5	250	350	5	2	5	11025
6	250	350	3	1	5	11025

For this section, cases 1 and 5 are chosen.

Case 1:

```
fc = 110;
fm = 220;
Io = 5;
tau = 2;
dur = 6;
fs = 11025;
x = bell ([fc, fm] , Io, tau, dur, fs);
soundsc (x, fs)
Ninc = 32;
tmax = dur*Ninc;
N = 1024;
figure (1)
spectrogram(x, hamming(N), N/8, N, fs, 'yaxis');
figure (2)
It = Io*bellenv(tau, dur, fs);
t = 0:1/fs: dur;
fi = fc - fm*sin(2*pi*fm*t).*It-(1/tau)*cos(2*pi*fm*t).*It;
plot (t, fi)
xlabel ('T(s)')
ylabel ('Instantaneous Frequency (Hz) #case 1 ')
figure (3)
plot(t,x)
xlabel ('T(s)')
ylabel ('Signal #case 1 ')
figure (4)
env = bellenv(tau, dur, fs);
plot (t, env)
xlabel ('T(s)')
ylabel ('Envelope signal A(t) #case 1 ')
figure (5)
plot (t(32975:33175), x(32975:33175))
xlabel ('T(s)')
ylabel ('x(t) #case 1 ')
```

Part (a)

The sound is made by multiple frequencies and it sounds like a bell. At the beginning it has more frequencies, but at the end, their power decreases except for f_c.

Part (b)

The fundamental frequency is gcd(f_c , f_m). Like the approach used in Section 3, a Taylor series expansion can be used to derive the frequencies that are present in the spectrogram. This signal is comprised of the frequencies

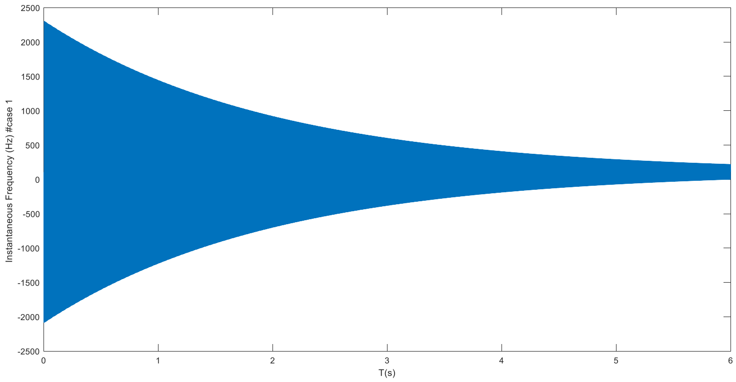
$$f = \pm f_c + k\, f_m = \pm 110 \text{ Hz} + k\; 220 \text{ Hz}$$

where k is integer. Therefore, the fundamental frequency is:

$$f_0 = \text{gcd}\,(f_c, f_m) = \text{gcd}\,(110 \text{ Hz}, 220 \text{ Hz}) = 110 \text{ Hz}$$

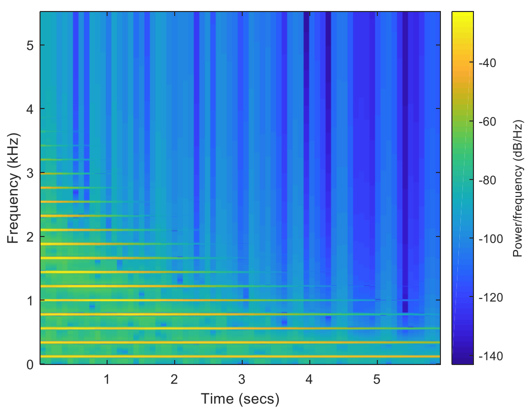
Part (c)

At first, higher number of frequencies can be detected, but with respect to the modulation index ($I(t)$), most of the frequencies will die out over time and only f_c coefficient remains constant.



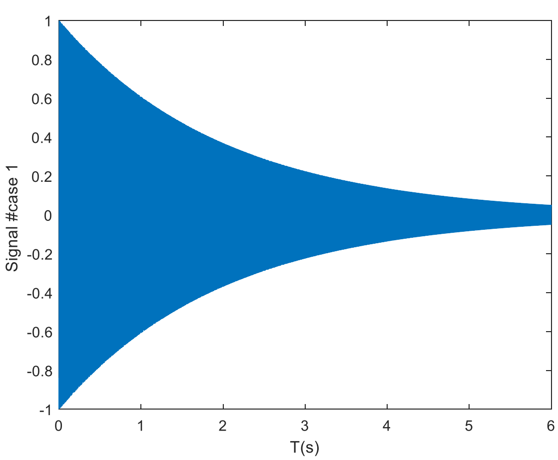
Part d:

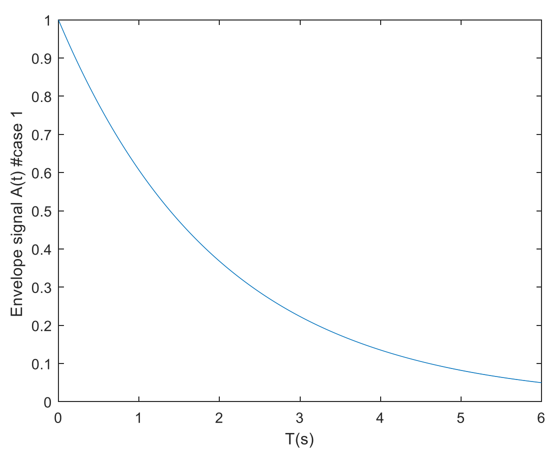
In part (b), the fundamental frequency has been calculated. In the spectrogram, all mentioned positive frequencies of the form $\pm f_c + k\, f_m$, are present. As the time passes, power of other frequencies decreases and only the f_c power, remains unchanged. Because the amplitude values of $I(t)$ decrease with respect to time, it shows its effect on the coefficient of other frequencies.



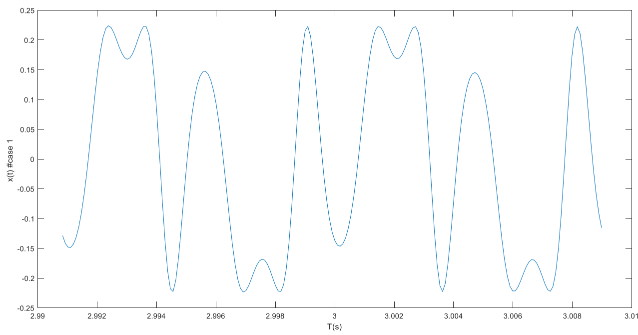
Part (e)

The changes in the signal shape match $A(t)$. For negative and positive amplitudes, the envelope function is visible, and that is due to the fact that sinusoid is multiplied by envelope function.





Part (f)



As shown in this figure, the signal has a high number of harmonics, and the fundamental frequency is 110 Hz.

Case 5:

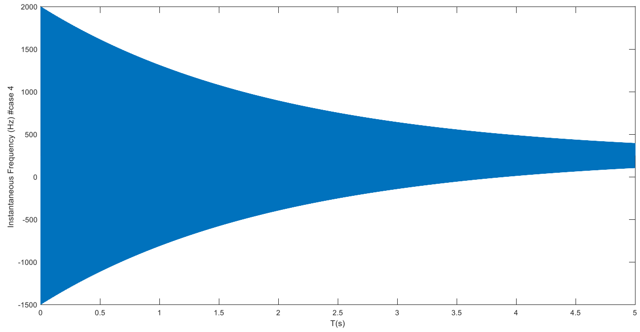
```
fc = 250;
fm = 350;
Io = 5;
tau = 2;
dur = 5;
fs = 11025;
x = bell ([fc, fm] , Io, tau, dur, fs);
soundsc (x, fs)
Ninc = 32;
tmax = dur*Ninc;
N = 1024;
figure (1)
spectrogram(x, hamming(N), N/8, N, fs, 'yaxis');
figure (2)
It = Io*bellenv(tau, dur, fs);
t = 0:1/fs: dur;
fi = fc - fm*sin(2*pi*fm*t).*It-(1/tau)*cos(2*pi*fm*t).*It;
plot (t, fi)
xlabel ('T(s)')
ylabel ('Instantaneous Frequency (Hz) #case 4 ')
figure (3)
plot(t,x)
xlabel ('T(s)')
ylabel ('Signal #case 5 ')
figure (4)
env = bellenv(tau, dur, fs);
plot (t, env)
xlabel ('T(s)')
ylabel ('Envelope signal A(t) #case 5 ')
figure (5)
plot (t(32975:33175), x(32975:33175))
xlabel ('T(s)')
ylabel ('x(t) #case 5 ')
```

Part (b)

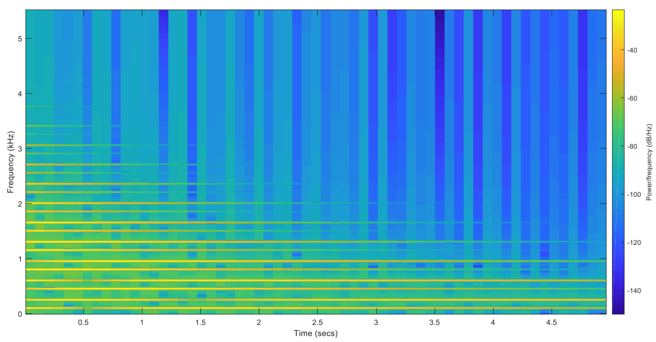
As mentiond in part (a), fundamental frequency is greatest common divisor of f_c and f_m:

f_0 = gcd (250 Hz, 350 Hz) = 50 Hz

Part (c)



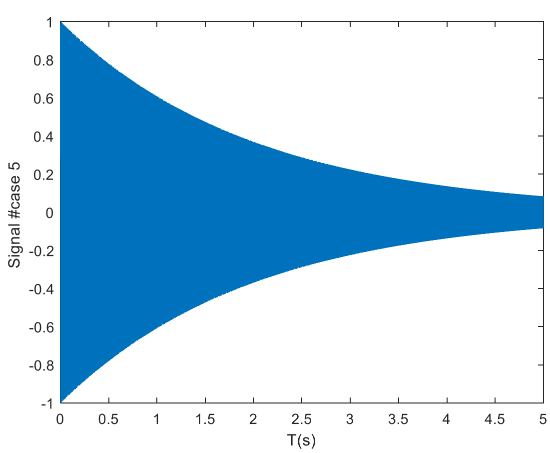
Part (d)

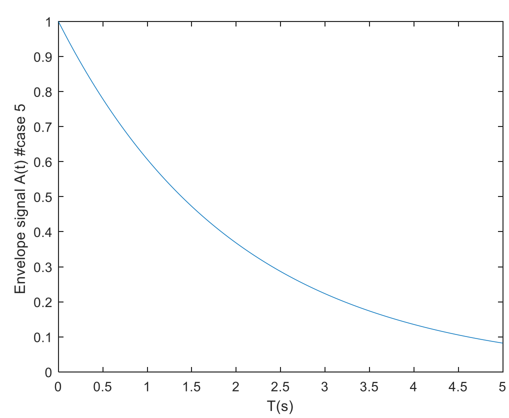


Here it can be seen that f_0 is different from f_c or f_m. In case 1 this could not be detected, because of the f_c: f_m ratio, e.g. 1:2, and gcd(f_c , f_m) = f_c . In this part, f_c:f_m ratio is 5:7 and gcd(f_c , f_m) is not equal to f_c. Also it can be seen that, f_c is the only frequency component that does not lose its power after 5 seconds.

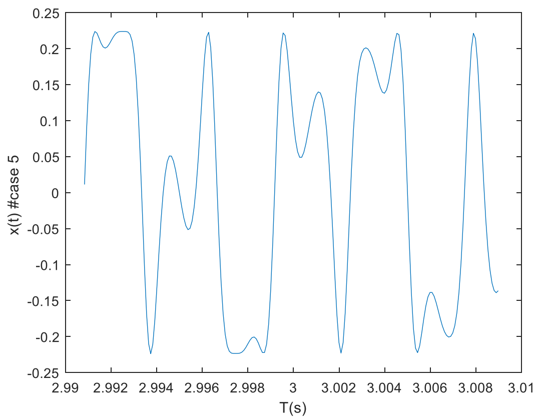
Part (e)

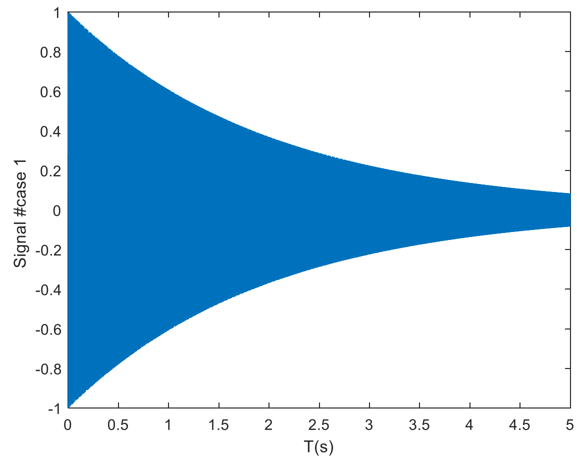
As in Case #1, signal is placed between $A(t)$ and $-A(t)$, which decreases the signal amplitude accordingly.





Part (f)





5.0 Conclusion (*18 points*)

In this mini-project, we performed a deep dive into the analysis of frequency modulated (FM) signals and their application to sound synthesis.

We first analyzed chirp signals. A chirp signal features a principal frequency that is increasing or decreasing linearly with time, and it is sometimes called a linear FM sweep. A chirp signal is used in ultrasound, radar, and active sonar as a way to gauge distances to objects in the environment; in audio systems to measure time and frequency responses; and in communication systems to measure distortion from transmitter to receiver. A chirp signal has a principal frequency with a narrow bandwidth.

Next, we analyzed frequency modulation in which the instantaneous frequency varied in a sinusoidal pattern with respect to time. Such signals also have a rich set of harmonics at integer multiples of the sinusoidal pattern frequency, which are offset by the carrier frequency and its negative value. Each harmonic has a narrow bandwidth, but span the entire frequency domain.

Finally, we used a more general form of frequency modulation in which the amplitude of the sinusoidal signal varies with time and the amplitude of the instantaneous frequency also varies with time. This means that as the instantaneous frequency varies in a sinusoidal pattern with respect to time, its strength is also varying with time. We used decaying exponentials for both amplitude functions to mimic the decaying amplitude and harmonic components with time when playing a bell by striking it with a hammer. Eventually, the principal frequency of the bell is the last sound heard. This modeling approach can be used to synthesize sounds for many woodwind and brass instruments.