EEE 321

Report: Signals and Systems Lab 1

Yigit Turkmen – 22102518

February 16, 2024

Introduction:

This report aims to analyze and understand the generation and perception of sinusoidal signals, especially within the context of musical sounds. By using MATLAB, the study focuses on the effects of frequency, amplitude, and phase on sound perception. If you can not see figures properly, please kindly zoom in.

Part 1: Fundamental Frequency and Harmonics

First, it is wanted that $x_1(t) = \sin(2\pi f_o t)$. by using this formula and taking f0 = 440Hz and with a sampling interval of 0.0001s the values of x1(t) should be stored in an array called "x1" with a duration of 3s, then we should plot the x1 vs t in MATLAB with a duration of 0.01s. Related code and plot are given as follows (Code 1 and Figure 1).

```
f0 = 440;
t = 0:0.0001:3;
x1 = sin(2*pi*f0*t);

figure(1)

plot(t,x1);
ylabel('x1');
xlabel('t')
xlim([0 0.01]);
title('x1 vs t')
```

Code 1

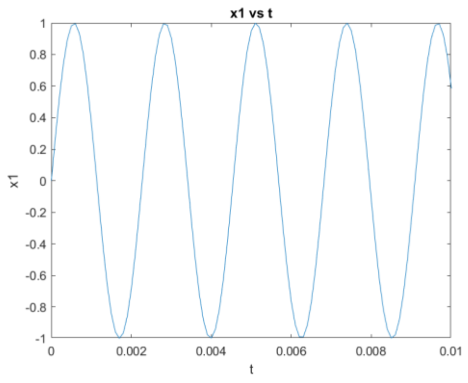
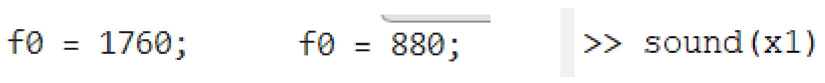


Figure 1: Plot of x1(t) vs t with a duration of 0.01s.

Now we use “sound(x1)” command to listen the wave and then by only changing f0 to 880Hz and 1760Hz with single line of codes (Code 2) we hear different sound where its pitch is getting higher with the increasing frequency.



```
f0 = 1760;        f0 = 880;        >> sound(x1)
```

Code 2

Question 1: What happens to the pitch of the sound as the frequency increases?

When the frequency of a sound wave is increased from 880 Hz to 1760 Hz, the perceived pitch of the sound rises. This is because the pitch that we hear is directly linked to the sound wave's frequency, with higher frequencies producing sounds of higher pitch.

Question 2: What happens to the pitch of the sound as the frequency increases? The calculation 440×3=1320 Hz is related to the formation of musical intervals and harmonics. Multiplying the fundamental frequency of A (440 Hz) by 3 gives us a frequency of 1320 Hz, which is an octave and a fifth above the original A note.

Question 3: Where does the C sharp come from in the major triad?

In the major triad, C sharp (C#) comes from a specific frequency calculation related to the note A (440 Hz). By multiplying the frequency of A (440 Hz) by 5, we approximate the frequency for C#, which is closely matched by multiplying the base frequency of C# (554 Hz) by 4, hence the relationship $440\ x\ 5\ \cong 554\ x\ 4$.

```
s = sin(2*pi*440*t) + sin(2*pi*554*t) + sin(2*pi*659*t);
figure(2)
plot(t,s);
ylabel('s');
xlabel('t')
title('s vs t')
```

Code 3: Code for the calculate s and plot it.

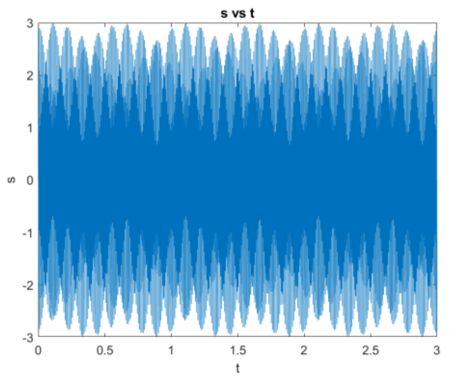


Figure 2: Plot of s vs t.

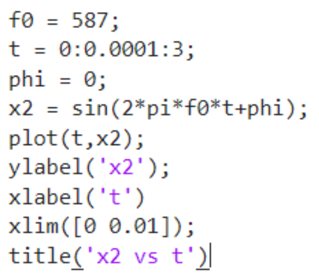
In Code 3, note that "t" is defined in the Code 1 which these two are in the same file; therefore, here we don't need to define "t" again. And while plotting (Figure 2) notice that there is no need for "`xlim([0 0.01])`" since we need to plot the duration of whole 3 seconds.

Question 4: What is special about this frequency combination?

The frequency combination of "s(t)" is special because it forms a major triad in music. The frequencies 440 Hz, 554 Hz, and 659 Hz correspond to the musical notes A, C#, and E, respectively. This combination is harmonically pleasing and is recognized as a major chord in Western music. The interval between A (440 Hz) and E (659 Hz), with a frequency ratio of 3/2, is known as a fifth. Further, the progression of multiplying frequencies by 3/2, and occasionally dividing by 2, traces the 12 notes of the musical scale, forming the circle of fifths. This circle underpins the notion of musical keys and scales. Therefore, this frequency combination is not only foundational to the construction of chords but also illustrates the harmonic structure that governs Western music theory.

Part 2: The Effect of Phase

In part “a” by $x_2(t) = \cos(2\pi f_o t + \phi)$ using this formula we need to store and calculate and plot the x2 vs t with a given f0 = 587Hz and phi = 0, note that “t” is the same as in part 1 (Code 4 and Figure 3).



```
f0 = 587;
t = 0:0.0001:3;
phi = 0;
x2 = sin(2*pi*f0*t+phi);
plot(t,x2);
ylabel('x2');
xlabel('t')
xlim([0 0.01]);
title('x2 vs t')
```

Code 4

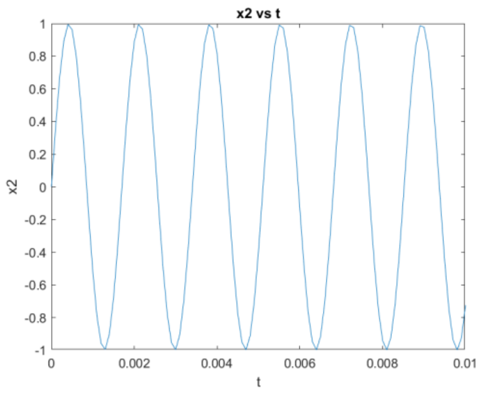


Figure 3: x2 vs t, phi = 0

Now in part b we will repeat this process for phi = pi/4, pi/2, pi3/4, pi. We need to just change the value of phi by single line of code in MATLAB like this: `phi = pi/2;` . See figures (4-7).

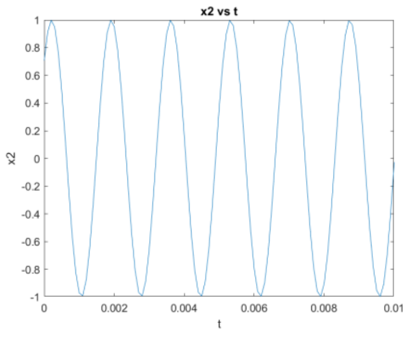


Figure 4: Phi = pi/4

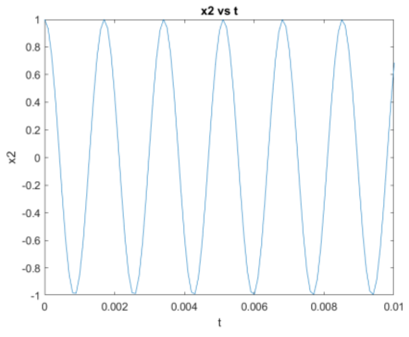


Figure 5: Phi = pi/2

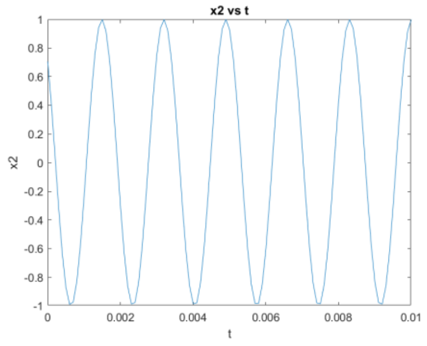


Figure 6: Phi = 3*pi/4

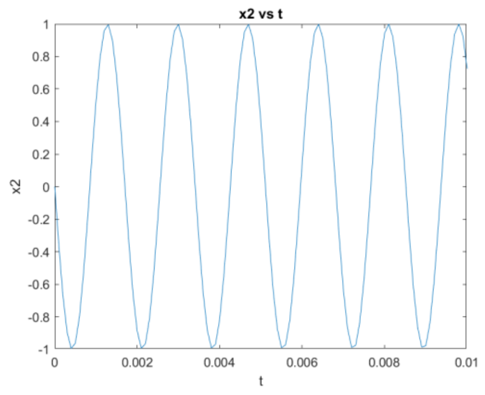


Figure 7: Phi = pi

Question 5: How does the volume of the sound that you hear change with φ? How does the pitch of the sound that you hear change with φ?

Since change of φ only affects phase of the function, and does not affect the frequency and amplitude, the volume and pitch of the sound don't change with the change of the φ.

Part 3: Sinusoid with Exponentially Decaying Envelope

In this part, we have the formula $x_3(t) = e^{-(a^2+2)t}\cos(2\pi f_o t).$ with initial values of a = 2 and f0 = 440Hz. By using the definition x3(t) and the values of a and f0 we need to compute and store the values in an array named x3, related codes and plot are given below (Code 5 and Figure 8) note that in fourth line ".*" is the code for the element-wise multiplication of arrays.

```
f0 = 440;
t = 0:0.0001:3;
a = 2;
x3 = exp(-(a^2+2)*t).*cos(2*pi*f0*t);
plot(t,x3);
ylabel('x3');
xlabel('t')
title('x3 vs t')
```

Code 5

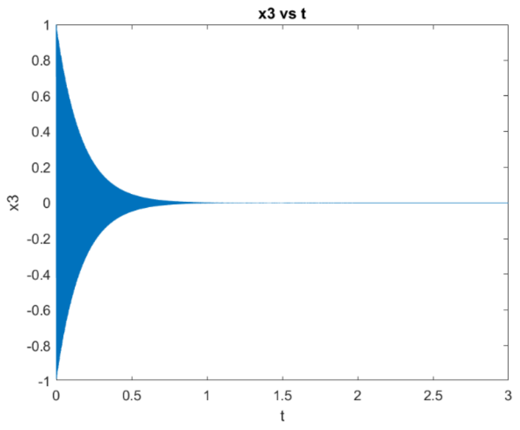


Figure 8: x3 vs t, a = 2

In the plot, it can be seen that as t increases, x3(t) tends to zero which means the signal diminishes as time passes. And this situation is not the case in x1(t) because the sound of x1(t) is stabil and not decreases as time passes which can be seen also by comparing the plots of these two signals.

Question 6: What is the effect of including the exponential term to the sound that you hear?

We have $e^{-(a^2+2)t}$ since -(a^2+2) is always negative, we have $1/\ (e^{(a^2+2)t})$ which means this term tends to zero as time increases with a constantly decreasing manner; therefore, including the exponential term will decrease the volume of the sound constantly, and eventually sound will not be heared.

Question 7: Which one of $x1(t)$ and $x3(t)$ resembles the sound produced by a piano more? Which one resembles that of a flute more?

X1 resembles flute sound more and x3 resembles piano sound more since x1 is stabil and x3 is decaying over time.

Now we change the value of a to 1 and 3 and repeat the process. See figures 9 and 10.

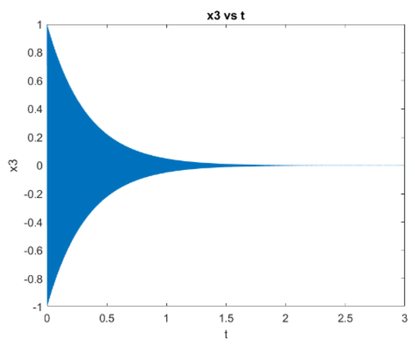


Figure 9: a = 1

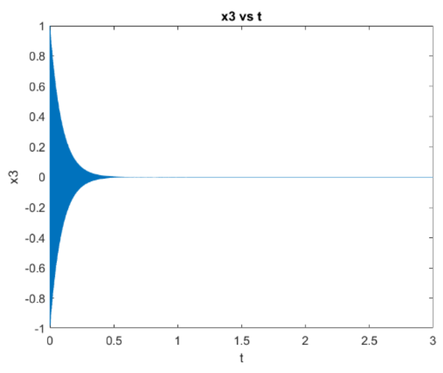


Figure 10: a = 3

So, as we make " a" bigger the exp term will quicklier vanish since the (a^2 + 2) is getting bigger and the opposite will apply for "a" is being less than the previous value. Therefore, the duration of the sound will be more less as "a" increases and the duration will increase as "a" decreases. Which is the case here the duration of a = 1 is bigger than a = 2, but the duration of a = 3 is less than a = 2 ones.

Question 8: How does the duration of the sound that you hear change as a increases?

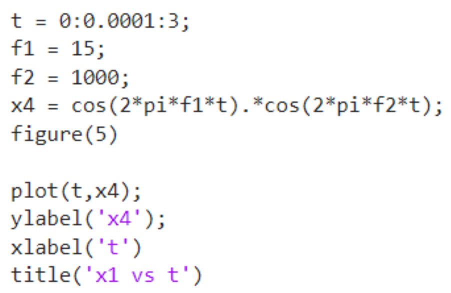
This question is answered above.

Question 9: Can you relate this signal model to a physical system that we discussed in class that produces sinusoidal signals?

Yes, RLC circuit is an example for this.

Part 4: Beat Notes and Amplitude Modulation

In this part we have $x4(t) = \cos(2\pi f_1 t)\cos(2\pi f_2 t)$ where *f1 << f2,* initially we have f1 = 10Hz and f2 = 1000Hz. We need to calculate the and store the values in an array named "x4" and then plot it see Figure 11 and Code 6. Note that the fourth line of the code is the single line code to calculate and store the values in x4.



```
t = 0:0.0001:3;
f1 = 15;
f2 = 1000;
x4 = cos(2*pi*f1*t).*cos(2*pi*f2*t);
figure(5)

plot(t,x4);
ylabel('x4');
xlabel('t')
title('x1 vs t')
```

Code 6

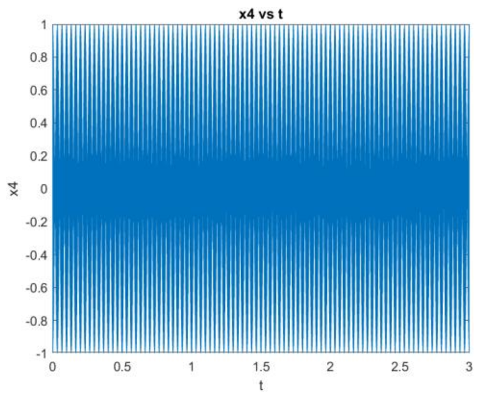


Figure 11: x4 vs t: f1 = 10Hz, f2 = 1000Hz

Question 10: What is the effect of the low-frequency cosine term cos $(2\pi f1t)$ on the sound that you hear?

Incorporating a low-frequency cosine term, such as cos$(2\pi f1t)$ into a sound synthesis formula influences the auditory perception through amplitude modulation (AM) of the signal. This modulation process induces a tremolo effect, characterized by a periodic fluctuation in the sound's volume or intensity, oscillating at the modulation frequency f1.

Question 11: Recompute **x4** for $f1$ = 5 Hz and $f1$ = 15 Hz. How does the sound that you hear change?

1. For f1=5Hz: The modulation frequency is relatively slow, which means the volume of the sound will fluctuate up and down at a rate of 5 cycles per second. This creates a gentle and more pronounced tremolo effect, where the change in loudness is easily perceptible and rhythmic. The slow modulation allows the ear to clearly discern the volume variations, adding a smooth, waving character to the sound. See figure 12.
2. For f1=15Hz: Increasing the modulation frequency to 15 Hz results in a faster rate of volume fluctuation. The tremolo effect becomes more rapid, with 15 cycles of volume

change occurring each second. This faster modulation can make the sound seem more vibrant or jittery, as the volume changes occur more quickly than at 5 Hz. The effect might be less individually discernible in terms of each volume peak and trough but will contribute to a sense of increased energy or tension in the sound. See figure 13.

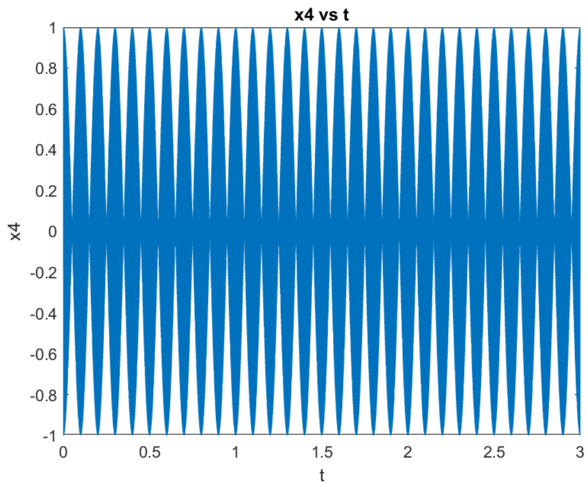


Figure 12: x4 vs t: f1 = 5Hz.

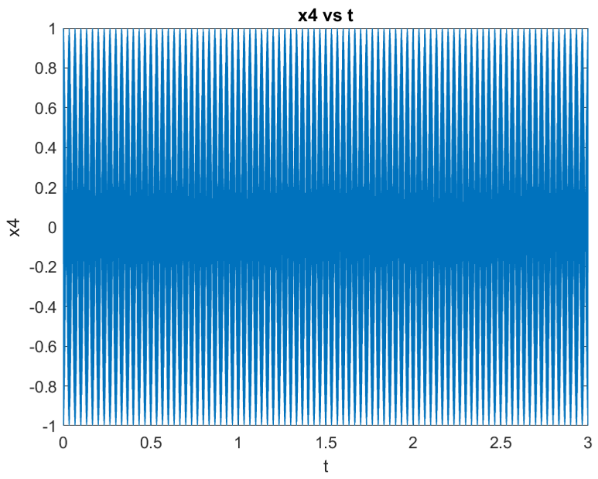
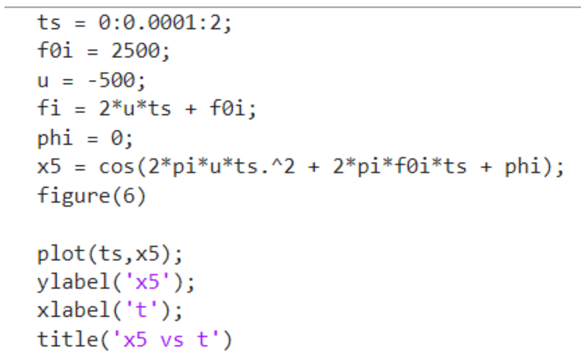


Figure 13: x4 vs t: f1 = 15 Hz.

Part 5: Chirp Signals and Frequency Modulation

In this part, we have $x5(t) = \cos(2\pi u t^2 + 2\pi f_0 t + phi)$ and initially f0 = 2500Hz by using the formula f = 2*u*t +f0, we know that duration is 2s; therefore, one can find the u simply using this equation which is: 500 = 2*u*2 + 2500 => u = -500. We also need the value of u for f0 = 500Hz and f_final = 2500Hz, again using the same intution: 2500 = 2*u*2 + 500 => u = 500. See Code 7 and Figures 14 to 17 for this part. Note that to change the value of u and f0 in MATLAB, only single lines of adjustment is sufficient in line 2-3 of the code. For scaling “xlim” function used in MATLAB.



```
ts = 0:0.0001:2;
f0i = 2500;
u = -500;
fi = 2*u*ts + f0i;
phi = 0;
x5 = cos(2*pi*u*ts.^2 + 2*pi*f0i*ts + phi);
figure(6)

plot(ts,x5);
ylabel('x5');
xlabel('t');
title('x5 vs t')
```

Code 7

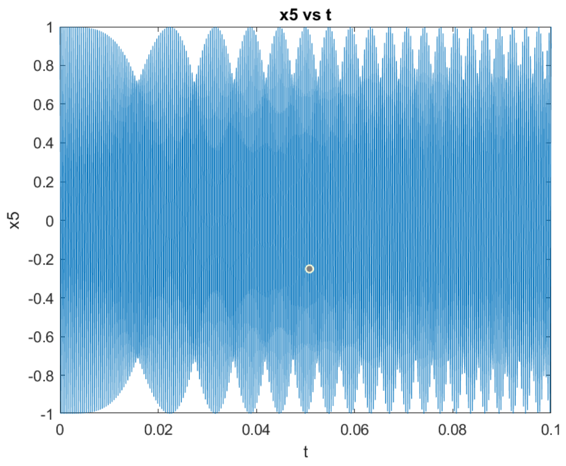


Figure 14: Scaled plot of x5 vs t: u = -500, f0 = 2500Hz

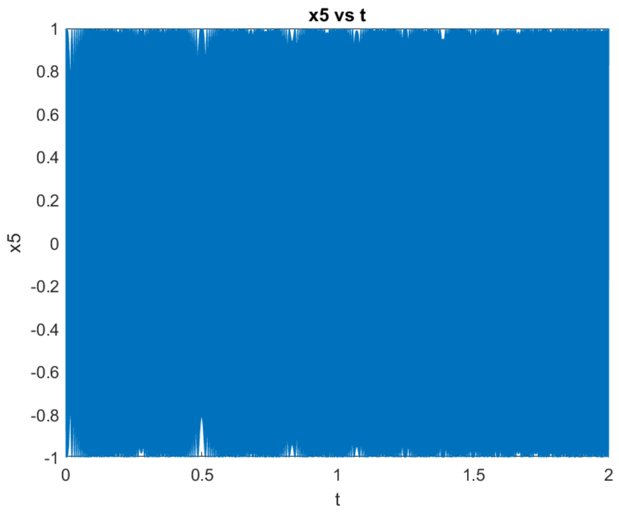


Figure 15: x5 vs t: u = -500, f0 =2500Hz.

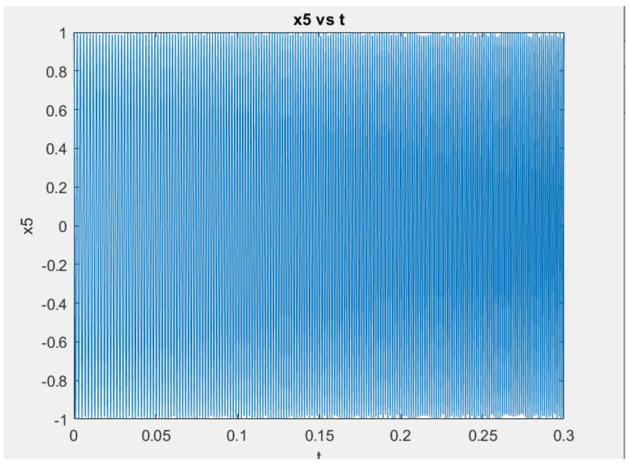


Figure 16: x5 vs t: u= 500, f0 = 500Hz (scaledversion)

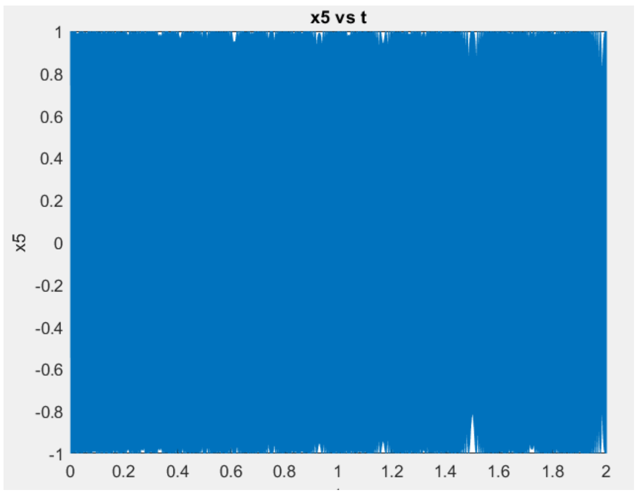


Figure 17: x5 vs t: u = 500, f0 = 500Hz.

Question 12: Listen to the chirp using the soundsc(.) function again. How is it different from the first chirp?

The difference between the u = -500 and u = 500 is while listening u = -500 chirp signal, the frequency of sound decreases by time and ;therefore, pitch of the sound decreases as the time passes. But while listening u = 500 chirp signal, the frequency of sound increases by time so, pitch of the sound increases as the time passes.

Question 13: What happens if you double the μ coefficient? What happens if you halve it?

If u gets doubled, the frequency will much quicklier increase or decrease rather than the previous one; therefore, the pitch of the sound decreased very quickly when i doubled the u. If u gets halved, the frequency will increase or decrease in a slower manner than the previous one; therefore, the pitch of the sound decreased not quickly as the first signal when i halved it.

A Chirp Puzzle:

Here we have: A total time duration of 3.0 s, with a sampling interval of Ts = 0:0001 s. The instantaneous frequency starts at 3000 Hz and ends at -2000 Hz. From the above formula

-2000 = 2*u*3 + 3000 => u = -2500/3. See Code 8 and Figure 18 and 19. Note that phi is defined in the above code.

```
ts1 = 0:0.0001:3;
f0i = 3000;
u = -2500/3;
fi2 = 2*u*ts1 + f0i;
x5_2 = cos(2*pi*u*ts1.^2 + 2*pi*f0i*ts1 + phi);
figure(7)

plot(ts1,x5_2);
ylabel('x5');
xlabel('t')
title('x5 vs t (A Chirp Puzzle)')
```

Code 8

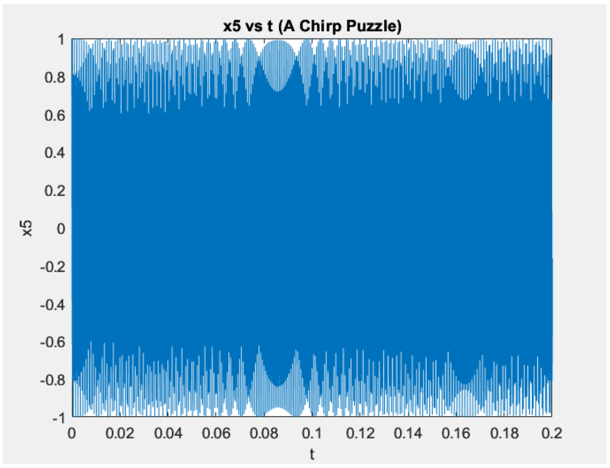


Figure 18: x5 vs t: u = -2500/3, f0 = 3000Hz (scaled)

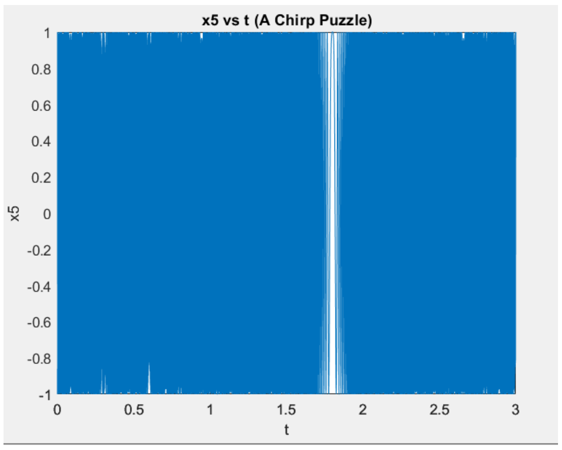


Figure 19: x5 vs t: u = -2500/3, f0 = 3000Hz.

Question 14: Does it chirp down, chirp up, or both? Use the theory of the frequency spectrum (with its positive and negative frequency components) to help explain what you hear.

Initially, as the frequency decreases, our ears perceive a pitch that goes down. Then, in the negative side as the absolute value of the frequency increases, we perceive the pitch going up. This change in frequency over time can be represented and analyzed using the frequency spectrum, considering both its positive and negative frequency components to fully understand the signal's behavior.

Part 6: Composing Music

For this part music file has been uploaded to moodle.

All codes for the Lab 1:

```
f0 = 440;
t = 0:0.0001:3;
x1 = sin(2*pi*f0*t);

figure(1)

plot(t,x1);
ylabel('x1');
xlabel('t')
xlim([0 0.01]);
title('x1 vs t')

s = sin(2*pi*440*t) + sin(2*pi*554*t) + sin(2*pi*659*t);
figure(2)
plot(t,s);
ylabel('s');
xlabel('t')
title('s vs t')
```

“lab1part1.m”

```
f0 = 587;
t = 0:0.0001:3;
phi = pi;
x2 = sin(2*pi*f0*t+phi);
plot(t,x2);
ylabel('x2');
xlabel('t')
xlim([0 0.01]);
title('x2 vs t')
```

“lab1part2.m”

```
f0 = 440;
t = 0:0.0001:3;
a = 3;
x3 = exp(-(a^2+2)*t).*cos(2*pi*f0*t);
plot(t,x3);
ylabel('x3');
xlabel('t')
title('x3 vs t')
```

“lab1part3.m”

```
t = 0:0.0001:3;
f1 = 5;
f2 = 1000;
x4 = cos(2*pi*f1*t).*cos(2*pi*f2*t);
figure(5)

plot(t,x4);
ylabel('x4');
xlabel('t')
title('x4 vs t')
x4_tri = 1/2*(cos(2*pi*f1*t+2*pi*f2*t)+cos(2*pi*f1*t-2*pi*f2*t));
figure(6)

plot(t,x4_tri);
ylabel('x4');
xlabel('t')
title('x4 vs t')
```

“lab1part4.m”

```
ts = 0:0.0001:2;
f0i = 500 ;
u = 500;
fi = 2*u*ts + f0i;
phi = 0;
x5 = cos(2*pi*u*ts.^2 + 2*pi*f0i*ts + phi);
figure(6)
plot(ts,x5);
ylabel('x5');
xlabel('t');
title('x5 vs t')
xlim([0 0.3]);

ts1 = 0:0.0001:3;
f0i = 3000;
u = -2500/3;
fi2 = 2*u*ts1 + f0i;
x5_2 = cos(2*pi*u*ts1.^2 + 2*pi*f0i*ts1 + phi);
figure(7)
plot(ts1,x5_2);
ylabel('x5');
xlabel('t')
title('x5 vs t (A Chirp Puzzle)')
```

“lab1part5.m”

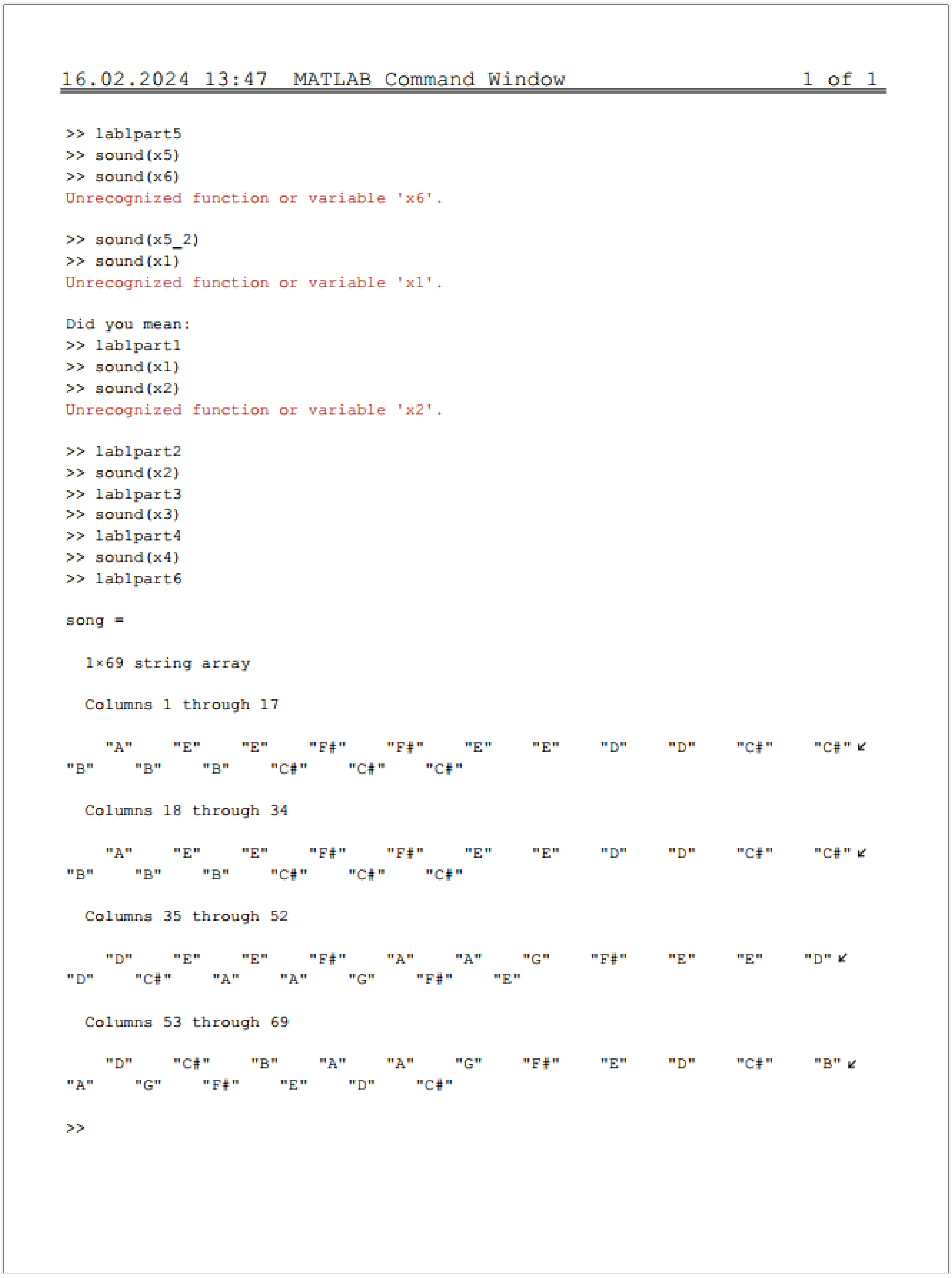
```
notename = ["A","A#", "B", "C", "C#", "D", "D#", "E","F", "F#", "G","G#"];
song = ["A", "E", "E", "F#", "F#", "E", "E", "D","D", "C#", "C#",
"B","B","B","C#","C#", "C#","A", "E", "E", "F#", "F#", "E", "E", "D","D", "C#",
"C#", "B","B","B","C#","C#", "C#","D", "E", "E", "F#", "A", "A", "G", "F#", "E",
"E", "D", "D", "C#", "A", "A", "G", "F#", "E", "D", "C#", "B", "A", "A", "G",
"F#", "E", "D", "C#", "B", "A", "G", "F#", "E", "D", "C#"]



for k1 = 1:length(song)
    idx = strcmp(song(k1), notename);
    songidx(k1) = find(idx);
end
dur = 0.3*8192;
songnote = [ ];
for k1 = 1:length(songidx)
    songnote = [songnote; [notecreate(songidx(k1),dur) zeros(1,75)]'];
end
soundsc(songnote, 8192)

function[note] = notecreate(frq_no, dur)
    note = sin(2*pi*[1:dur]/8192*(440*2.^((frq_no-1)/12)));
end
```

“lab1part6.m”



```
16.02.2024 13:47  MATLAB Command Window                        1 of 1

>> lab1part5
>> sound(x5)
>> sound(x6)
Unrecognized function or variable 'x6'.

>> sound(x5_2)
>> sound(x1)
Unrecognized function or variable 'x1'.

Did you mean:
>> lab1part1
>> sound(x1)
>> sound(x2)
Unrecognized function or variable 'x2'.

>> lab1part2
>> sound(x2)
>> lab1part3
>> sound(x3)
>> lab1part4
>> sound(x4)
>> lab1part6

song =

  1×69 string array

  Columns 1 through 17

    "A"    "E"    "E"    "F#"    "F#"    "E"    "E"    "D"    "D"    "C#"    "C#" ↙
"B"    "B"    "B"    "C#"    "C#"    "C#"

  Columns 18 through 34

    "A"    "E"    "E"    "F#"    "F#"    "E"    "E"    "D"    "D"    "C#"    "C#" ↙
"B"    "B"    "B"    "C#"    "C#"    "C#"

  Columns 35 through 52

    "D"    "E"    "E"    "F#"    "A"    "A"    "G"    "F#"    "E"    "E"    "D" ↙
"D"    "C#"    "A"    "A"    "G"    "F#"    "E"

  Columns 53 through 69

    "D"    "C#"    "B"    "A"    "A"    "G"    "F#"    "E"    "D"    "C#"    "B" ↙
"A"    "G"    "F#"    "E"    "D"    "C#"

>>
```

“Commandwindow.pdf”