Name(s):

Introduction

The Fourier transform can transform a signal from its time domain to a frequency domain, can decompose a signal, and can therefore help us manipulate and analyze signals. This is helpful in audio editing and image decomposition. This assignment will focus on the fourier transform and its applications in audio processing.

Objective

- Understand the use of fourier transforms in audio and be able to apply the fourier transform to a signal.
- Create various signals using fourier transformations to imitate music notes and reconstruct a small portion of the song “Take Me Out to the Ball Game”.
- Create an audio effect using the Fourier transform properties.

Other Information

- This assignment utilizes MATLAB script. It is *highly encouraged* to use this instead of any other language or compiler.
- All code and tables can be found at the end of the PDF or can be found in the textbook *Signals and Systems: A MATLAB Integrated Approach* by Oktay Alkin.
- Along with this instruction sheet, you will be provided an MP3 file. Please download freemusicbg.com-Ballgame and place this file in your MATLAB folder.
- *If your team uses AI to aid in any code or calculations, no points will be deducted so long as you **cite** where it is used and create a short comment about why you used it and what you learned from it.*

Part 1 – The Fourier Transform

Pure toned musical notes can be represented as the following continuous time function:

$$y(t) = Asin(2\pi ft + \phi)$$

$A = amplitude$
$f = fundamental\ frequency$
$t = time$
$\phi = phase\ offset$

The amplitude affects the volume of the note and the frequency affects the tone of the note. In real life, notes created by analog instruments are much more complicated because of overtones and harmonics created by the instrument. For now, we will stick with the pure tone signal.

To manipulate and deconstruct signals, it is easier to work in the frequency domain. The general analysis equation for a forward transform is given by the following equation:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$$

Alkin 307. Figure 4.130

The signal $y(t) = sin(523.26\pi t)$ is a pure tone signal, representing the note C4 (a C at the 4th octave). The fundamental frequency is 261.63 Hz, the amplitude is 1, and there is no phase offset.

1) Find the fourier transform to transform this signal to the frequency domain. You may use any method and/or reference Table 4.4 in our *Signals and Systems: A MATLAB Integrated Approach* on page 366. You may also use other known identities, such as Euler's law to find the fourier transform. List the method you use and show your work:

[Answer Here]

Part 2 – Creating a MATLAB Script

Download the MP3 file freemusicbg.com-Ballgame and place this in your MATLAB folder.

Open MATLAB and create a new script. Copy Script 1 (found in the Appendix of this assignment) into the Editor. Run the script.

-> Explanation: Script 1 runs the first 5 seconds of a version of the song "Take Me Out to the Ball Game" which we will call "Ballgame". It then plots the signal in the time domain. Applying a fast fourier transform, which is used specifically for discrete signals. Discrete fourier transforms are typically used when modifying on technology since all computers process discrete signals and not continuous signals.

1) Briefly explain in your own words what Figure 1 represents when you run Script 1 and what information can be learned from them.

[Answer Here]

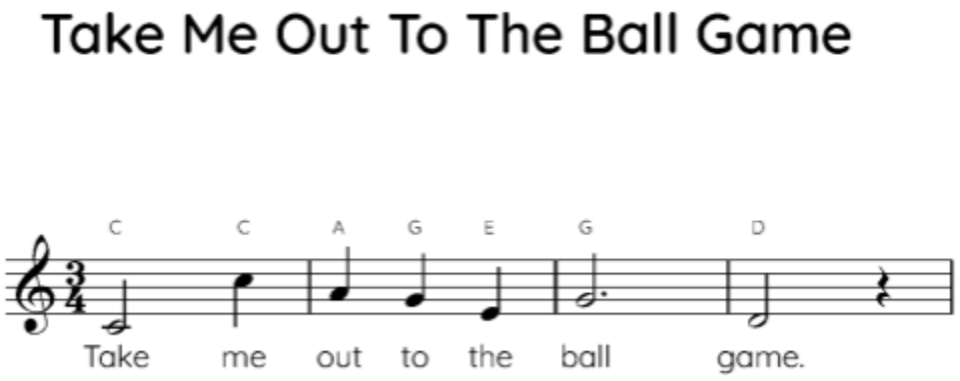
2) Think about the process you took in Part 1 to do a fourier transformation. Create a script that will automate the process given *any* frequency input (the user should be prompted to input a frequency). Assume the base function is y(t)=sin(2πft). The output should include a text output of the full signal and a graph of the signal in both the frequency and time domain. YOUR CODE SHOULD INCLUDE COMMENTS. Copy and paste your code below:

[Answer Here]

Create an audio output by using the MATLAB function ‘audiolayer(y,fs)’. Use Script 2 in the Appendix as a sample code.

-> You should now be able to input any frequency and create a note in MATLAB, as well as visualize it in its frequency domain.

Additionally, you can use Table 1 in the Appendix to find the specific frequencies of specific tones. We recommend updating your code to take the input of a note (ex. C4) and recognize this requires a fundamental frequency of 261.63 Hz. For ‘Ballgame’ you will need 7 notes:



C4, C5, A4, G4, E4, D4

Add to your script code that will play the given notes (no bonus points for matching the rhythm, but it's fun if you do).

3) Add your final code to a new page under the Appendix and label it “[Last Names] Script 1 - Part 2”. Screenshot and label your MATLAB figures below:

[Answer Here]

End Goals:

- Your script should be able to take a frequency input into a pure tone signal, do a fourier transform and output the fourier transform as text, as a graph, and as an audio output.

Part 3 – Further Exploration/Audio Effects

The basic Fourier transform allows us to convert a signal from the frequency or time domain. Once a signal is in the frequency domain, we can create many audio effects such as echo, reverb, and chorus effects to name a few.

The echo effect can be done by time shifting a signal. For example:

$$y(t) = x(t) + ax(t - \tau)$$

$a = amplitude/attenuation\ (typically\ a\ smaller\ amplitude\ than\ the\ original\ signal)$
$\tau = the\ delay\ (time\ shift)$

We can apply the time delay property of the Fourier transform as such:

$$x(t - \tau) \leftrightarrow X(\omega)e^{-j\omega\tau}$$

Alkin 366. From Table 4.4

[Optional] Create a MATLAB script that creates an echo effect. The more time shifts you include, the more "echoes" you will have.

[Mandatory] A chorus effect utilizes several types of fourier properties, including time shifting, frequency modulation, linearity, and amplitude scaling (all of which can be found in Table 4.4 from the textbook). Watch the following short for an example: The Chorus Effect Explained in 30 Seconds

1) Briefly describe what each of the listed properties of the Fourier transform will do to a signal, and how that might sound (ex. "The time delay plays the note again quieter creating an echo effect.").

[Answer Here]

2) Use the pure tone signal, $y(t) = sin(523.26\pi t)$ to create a variation of the chorus effect. You must use the time shifting property, and at least one other property of your choice. Before creating a script, show the transformation properties you used and your work to find the fourier transform of the signal:

[Answer Here]

3) Create a script that will do these transformations and apply these properties, and additionally play the effect. Make at least one graph that shows an output of a transformation completed and take a screenshot and list it here (please label your graph!):

[Answer Here]

4) Look at Q1 of this part again. Were your descriptions accurate to what you hear in your simulation?

[Answer Here]

Be sure to list your final code in a new page under the Appendix and label it “[Last names] Script 2 - Part 3” YOUR CODE SHOULD INCLUDE COMMENTS.

Appendix

Script 1 - Coleman, Grix

```
clear;
clc;
%opening an audio file
aud = audioinfo('freemusicbg.com-Ballgame.mp3');
%read audio file
[y,Fs] = audioread('freemusicbg.com-Ballgame.mp3');
%sample of first 5 sec
num_secs = 5;
num_samps = num_secs*Fs;
f = y(1:num_samps,1);
N = length(f);
t = linspace(0, num_secs, num_samps);
%plot the audio (time domain)
figure(1);
subplot(2,1,1);
plot(t, f);
title('Audio (Time Domain)');
xlabel('Time (s)');
ylabel('Amplitude');
hold on;
%diplay audio info in command window
disp(aud);
%use fast fourier transform
aud_fft = fft(f);
% Normalize the FFT output for better visualization
aud_fft = aud_fft / length(f);
%plot transform
figure(1);
subplot(2,1,2);
plot(abs(aud_fft(:,1)));
xlim([0 Fs/2]);
title('FFT (Magnitude Spectrum)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
hold on;
```

Script 2 - Coleman, Grix

```
%construct the approx notes
fs = 1e4; % sampling frequency
t = 1:1/fs:5; % time signal
freq = 261.6; % note frequency in Hz (C, octave 4)
y = sin(2*pi*freq*t); % create sine wave (don't get rid of)
player = audioplayer(y, fs); % create audio player object
play(player); % play sound
figure(2);
subplot(3,4,1:4);
plot(y);
title('C4');
xlabel('Time (s)');
ylabel('Amplitude');
hold on;
```

Table 1

NOTE FREQUENCY CHART | HEROIC AUDIO

	Octave 0	Octave 1	Octave 2	Octave 3	Octave 4	Octave 5	Octave 6	Octave 7	Octave 8	Octave 9	Octave 10
C	16.35	32.70	65.41	130.81	261.63	523.25	1046.50	2093.00	4186.01	8372.02	16744.04
C#	17.32	34.65	69.30	138.59	277.18	554.37	1108.73	2217.46	4434.92	8869.84	17739.69
D	18.35	36.71	73.42	146.83	293.66	587.33	1174.66	2349.32	4698.64	9397.27	18794.55
D#	19.45	38.89	77.78	155.56	311.13	622.25	1244.51	2489.02	4978.03	9956.06	19912.13
E	20.60	41.20	82.41	164.81	329.63	659.26	1318.51	2637.02	5274.04	10548.08	
F	21.83	43.65	87.31	174.61	349.23	698.46	1396.91	2793.83	5587.65	11175.30	
F#	23.12	46.25	92.50	185.00	369.99	739.99	1479.98	2959.96	5919.91	11839.82	
G	24.50	49.00	98.00	196.00	392.00	783.99	1567.98	3135.96	6271.93	12543.86	
G#	25.96	51.91	103.83	207.65	415.30	830.61	1661.22	3322.44	6644.88	13289.75	
A	27.50	55.00	110.00	220.00	440.00	880.00	1760.00	3520.00	7040.00	14080.00	
A#	29.14	58.27	116.54	233.08	466.16	932.33	1864.66	3729.31	7458.62	14917.24	
B	30.87	61.74	123.47	246.94	493.88	987.77	1975.53	3951.07	7902.13	15804.26	

Grading Rubric

Task	1 point	2 points	3 points	4 points	5 points
Part 1 – The Fourier Transform	The Fourier transform is not computed correctly and little to no work is shown.	The Fourier transform is not computed correctly with some work shown and methods listed.	The Fourier transform is computed with few errors and most work shown and methods listed.	The Fourier transform is computed with minor error(s) with all work shown and methods listed.	The Fourier transform is correctly computed with all work shown and methods listed.
Part 2 – Creating a Program (Q1-Q2)	Q1 response lacks depth, the code is missing various parts/the team did not follow the instructions.	Q1 response lacks depth, the code might run, is not well organized, has issues with the outputs requested, or does not output everything requested.	Satisfactory response to Q1, the code runs well but is not organized, there are minor issues in the outputs requested.	Satisfactory response to Q1, the code runs and is well organized, there are very minor issues in the outputs requested.	Thoughtful response to Q1, the code runs and is well organized, takes a frequency input, and it outputs the signal as text and the correct graphs in the time and frequency domains.
Part 2 – Creating a Program (Q3 - the final script)	The script has many errors and is missing parts of Q2 or Q3. The script may not work.	The script has many errors or is missing parts of Q2 or Q3.	The script has a few errors, but does output everything requested in Q2 and Q3.	The script is well organized and incorporates everything from Q2 and Q3, but may have minor errors.	The script is well organized and outputs everything from Q2, in addition to playing the final melody.
Part 3 – Further Exploration (Q1 and Q4)	Limited response, Q4 may not show comparison between predictions	Responses lack thoughtfulness/depth. Q4 may not show comparison	Responses lack thoughtfulness or reflection on the differences/similarities of	Responses are thoughtful and show reflection of the prediction and simulation.	Responses are thorough and thoughtful, and show reflection of the prediction

Task	1 point	2 points	3 points	4 points	5 points
	and the simulation.	between prediction and simulation.	the simulation.		vs the simulation.
Part 3 – Further Exploration (Q2)	Only one property was explored or the students did not show all work, or had many errors in their transformation.	The time shifting property and at least one other property is explored. The students did not show all work or had errors in their transformations.	The time shifting property and at least one other property is explored. The students showed most work and had some errors in their transformations.	The time shifting property and at least one other property is explored. The students show all work and may have minor errors in their transformations.	The time shifting property and at least one other property is explored. The students show all work and have accurate transformation results.
Part 3 – Further Exploration (Q3)	The script may not run, may not play the effect and has many errors.	The script runs but may not play the effect and has many errors.	The script is well organized but does not play the effect or has errors.	The script is well organized, plays the effect, makes at least one graph, but may have minor errors.	The script is well organized, plays the effect*, makes at least one graph to show the output.
Total					**__/30 pts**

***The actual audio of the effects in Q3 will differ from group to group. As long as some sort of effect can be heard, and it aligns with the chosen properties, this counts as credit for this stipulation.**