ENGIN 112: Homework 2

Due date: 9/30/22 5:00pm

Please submit your answers via Gradescope. You can write your answers electronically or by hand and submit a scan or photo.

Question 1

(Signal size and compression): We discussed MP3 as an example of data compression applied to audio signals. Data compression techniques are also commonly applied to images (e.g., JPEG) and video signals (e.g., MPEG-4). These are similar to MP3 in that they make use of frequency components.

(a) Consider a digital camera that you commonly use (e.g., one on your phone). How many pixels are in a photo taken by the camera?

(b) Color pictures have three color components (RGB – red, green, blue). Each component is typically encoded with 8 bits, so we have 24 bits (i.e., 3 bytes) per pixel. With no compression coding, how much memory would be required to store a picture taken by your camera?

(c) Consider one stored photo from your camera. How much memory is actually used?

(d) The **code rate** for a compressed signal is defined as the ratio {number of bits used to store the uncompressed signal}/{number of bits used to store the compressed signal}. What is the code rate for the photo of part (c)? How does that compare to the code rate of the version of MP3 discussed in class?

Question 2

(Signals, sinusoids, Fourier and MATLAB): This problem is meant to get you started using MATLAB, which is a very useful software package for computation, plotting, signal processing, and system simulation. It is used in several ECE courses. UMass Amherst provides a campus-wide license with unlimited use of MATLAB, Simulink, and online learning tools for all students, faculty, staff, and researchers, on and off campus, on any device. Find out more and download what you need from the UMass Amherst MATLAB Portal.

As noted in class, every signal can be written as a sum of sinusoids having different frequencies. In this problem we will use MATLAB to see how we can add different sinusoids to create a signal that is not itself a sinusoid. First, since MATLAB is a program it has to use digital representations for signals. To get a representation that looks "almost analog" we'll use very closely spaced samples. In MATLAB, define the set of sample times with the command

```
t=[0:0.001:3.999];
```

This creates a set of times starting at 0 and ending at 3.999, with a spacing of 0.001 between samples (so, the sample times are 0, 0.001, 0.002, … , 3.999). (Note that using a semicolon at the end of a MATLAB command prevents the result from being printed to the screen; if you forget the semicolon all 4000 values of t will appear on the screen.)

We will use sinusoids (in this problem, sine functions) with frequencies $0, 0.5, 1.5, 2.5, \ldots, 49.5$ Hz (a total of 51 frequencies). MATLAB represents sampled signals as either row or column vectors. We'll represent our sinusoids as row vectors, and store them as rows of a matrix S. First note that frequency = 0 means that the signal is constant (no oscillations). To create a signal having the same length as t that is constant with amplitude = 1, we can use the command

```
s0=ones(size(t));
```

We can then create the matrix S with the following commands:

```
S=s0;
for k=2:51
S=[S; sin(2*pi*(k-1.5)*t)];
end
```

(These commands put s0 as the first row of S, and then build the full S by adding a new row at each step in the "for" loop. The kth row is filled with the sine function having frequency (k-1.5) Hz for k= 2,…,51.)

(a) To see that you have the right sinusoids stored in S, try plotting a few rows. For example, to plot the 5th row of S in red and the 9th row in blue on the same figure you can use the command

```
plot(t,S(5,:),'r',t,S(9,:),'b')
```

(The notation S(5,:) means all the points in row 5 of S.) You can adjust the axis limits with the axis(…) command – for example, to have the plot cover x-axis values 1 to 2 and y-axis values -1.2 to 1.2, after the plot command put the command

```
axis([1,2,-1.2,1.2])
```

You can also label the axes and put a legend on the plot– for example:

```
xlabel('time')
```

```
ylabel('amplitude')
legend('row 5 of S','row 9 of S')
```

Print your plot and attach it to your homework.

(*Note:* To get a detailed description of how a MATLAB command works, including the different options available in the command, just type

help *command*

For example, to learn more about what you can do with the plot command, type

help plot)

(b) Now we will create a sum of sinusoids having the form

$$x(t)=1+\frac{4}{\pi}\sin(\pi t)+\frac{4}{3\pi}\sin(3\pi t)+\frac{4}{5\pi}\sin(5\pi t)+\cdots$$

(so, aside from the first term (that is, the $f=0$ term), the sinusoids in the sum have the form $\frac{2}{(k-1.5)\pi}\sin(2\pi[k-1.5]t)$ for $k=2,3,4,\ldots,51$). We will make a matrix X where the first row of X is the first term in $x(t)$, the second row is the sum of the first two terms, the third row is the sum of the first three terms, etc. Note that the sinusoids are stored as rows of S, so we just need to sum the rows of S multiplied by the appropriate constants. To do this we can use the commands

```
X=S(1,:);
for k=2:51
Ak=2/((k-1.5)*pi);
X=[X;X(k-1,:)+Ak*S(k,:)];
end
```

(This set of commands puts the first row of S as the first row of X. In the "for" loop, we set Ak to be the coefficient of the kth term in the equation for *x*(*t*); then we create the kth row of X by adding the (k-1)th row to Ak times the kth sinusoid stored in S.)

Plot the sums of the first 4 terms, the first 12 terms, and all 51 terms in $x(t)$. Attach the plots to your homework. (*Note:* You can plot the three signals separately; plot all three signals on one graph using different colors (like in the plot of part (a)); or plot them as separate rows in a single plot using the "subplot" command – for example, the commands:

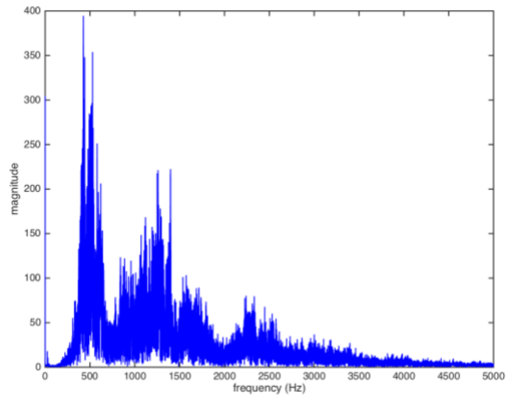
```
subplot(3,1,1),plot(t,X(4,:),'r')
subplot(3,1,2),plot(t,X(12,:),'m')
subplot(3,1,3),plot(t,X(51,:),'b')
```

plot X(4,:) in red on the first row, X(12,:) in magenta on the second row, and X(51,:) in blue on the third row. You can also scale axes, add axis labels and legends, etc., to any of the subplots as described in part (a).)

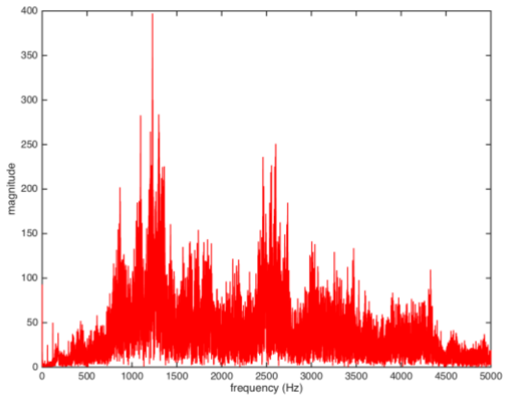
(c) Draw a plot (by hand or with MATLAB) of what you think $x(t)$ will look like if we sum a very large number of sinusoidal terms.

Question 3

(Spectrum) (a) The audio signals yay.wav and boo.wav have been placed on Moodle. Consider the two spectra shown at the top of the next page:



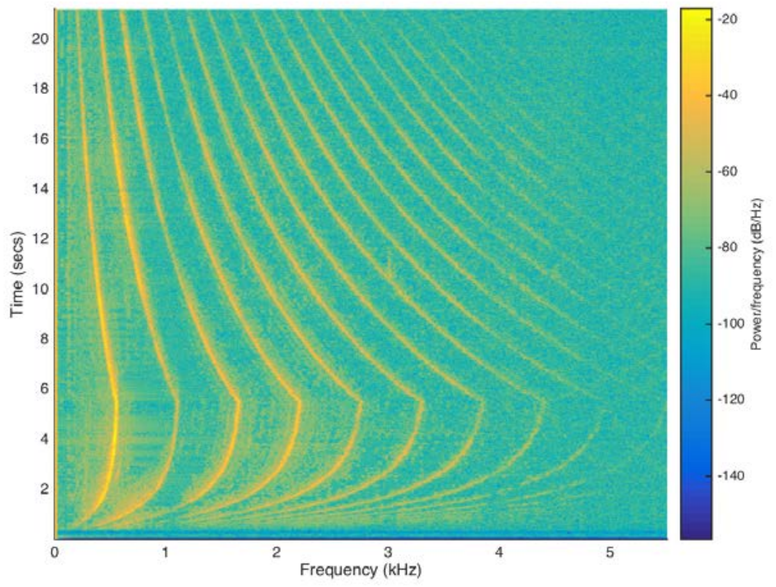
Spectrum 1



Spectrum 2

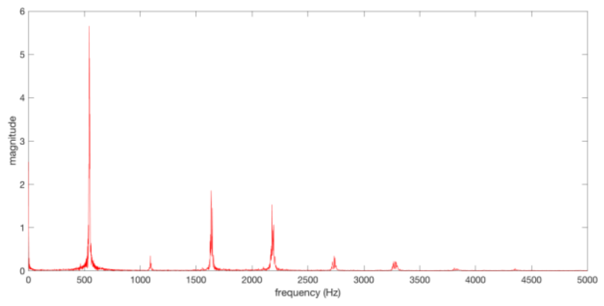
Which spectrum do you think goes with “yay”, and which with “boo”? Explain your reasoning.

(b) Another audio signal, siren.wav, has also been placed on Moodle. Its spectrogram is shown below. (Brighter color in the spectrogram indicates higher magnitude.)

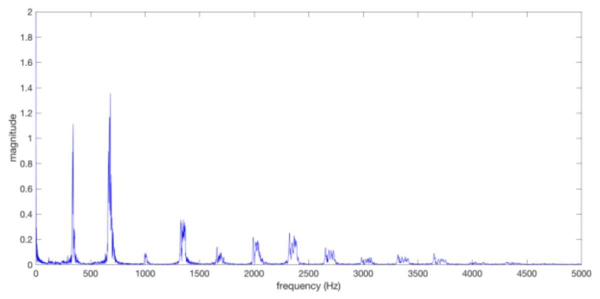


Siren signal spectrogram

Spectra for clips taken at two different times from the siren signal are shown on the next page (each clip was about 0.5 sec. long). Comparing each spectrum to the full spectrogram, from which time in the full signal do you think each clip was taken (e.g., around 2 sec. into the signal, around 10 sec. into the signal, etc.)? Explain the reasoning behind your answers.



Spectrum of clip 1



Spectrum of clip 2

Question 4

(Sampling and Quantization): The Federal Communications Commission (FCC) requires that the audio signals transmitted over FM radio[1] have a highest frequency of 15 kHz.

[1] https://en.wikipedia.org/wiki/FM_broadcasting

(a) Find the minimum sampling rate that we need to avoid aliasing if we sample audio signals used in FM radio.

(b) Say that we want to quantize the samples from part (a). How many bits/sample do we need to use if we want SQNR > 80 dB?

(c) Say that we use a sampling rate of 35 kHz and that we quantize to 12 bits/sample. Stereo FM requires two signals (the right and left channel signals) for each transmitted song. How many bytes would be needed to store a 4-minute song?

Question 5

(Aliasing) : Say that a signal $x(t)$ has its highest (Nyquist) frequency at f_H Hz. We know that if we use a sampling rate $f_S > 2f_H$ we will not have aliasing. Now suppose that we use a sampling rate that is too low – say, $f_S < 2f_H$. Then aliasing works like this: say that $x(t)$ has a frequency component at f_1 where $f_S/2 < f_1 < f_H$. The spectrum of the sampled signal will now have a component appearing at the **aliased frequency** $f_{1,a} = f_S - f_1$. $f_{1,a}$ is an **alias**

Example: Suppose $x(t)$ has its highest (Nyquist) frequency of $f_H = 10$kHz, and we sample at the rate $f_S = 16$kHz. Consider the spectral component of $x(t)$ at $f_1 = 9$kHz. This component of the spectrum of $x(t)$ now appears in the spectrum of the sampled signal as the **aliased frequency** $f_{1,a} = f_S - f_1 = 16 - 9 = 7$kHz. This 7kHz signal is an **alias.**

(a) Recall that in the Module 2 slides, we highlighted aliased components in the spectrum of the word “hello” when it was sampled at 5.512 kHz. In what frequency range should those components have appeared (that is, what was the range of those frequency components in the original signal)?

(b) If a signal $x(t)$ has a highest frequency of f_H Hz, at what sampling rate will the component at f_H Hz appear at 0 Hz (that is, look like a constant) in the interpolated signal?

(c) You can see an aliased video of a flying helicopter at

https://www.youtube.com/watch?v=R-IVw8OKjvQ

A typical helicopter rotor speed is 600 revolutions per minute. Assuming that speed, what was the frame rate (in frames/sec.) of the video?

Question 6

(Human hearing and audio signal processing): The human auditory (hearing) system shares several features with the audio signal processing systems we discussed in class. In particular, the ear's response to an incoming sound signal is to generate something similar to a spectrogram of the signal – that is, it divides the signal into components at different frequencies, and measures how the sizes of those components vary with time. The brain uses the pattern of magnitudes vs. time and frequency (that is, the "spectrogram" generated by the ear) to recognize the sound (although exactly how recognition is performed is not fully known). A good discussion of the processing involved in human hearing can be found at the web site

https://auditoryneuroscience.com/

(The site is run by the University of Oxford in England.) The site contains several interesting videos and discussions of how we create speech, how different parts of the ear work, how hearing changes as a person grows older, etc. It also has an online spectrogram - follow the link on the main page to "Speaking", or go directly to

https://auditoryneuroscience.com/acoustics/spectrogram

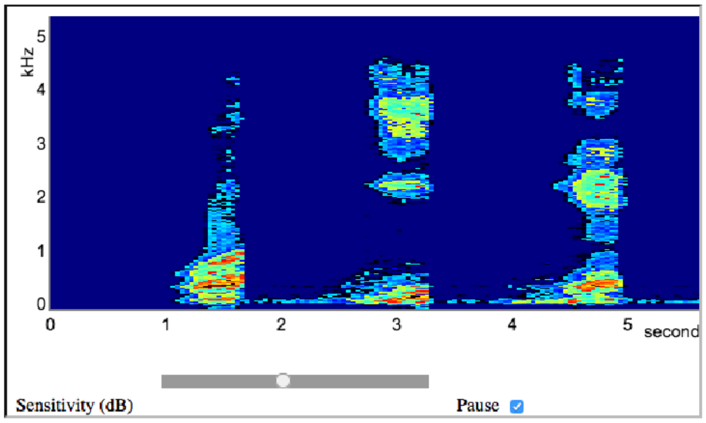
This generates a spectrogram of sounds spoken into your computer's microphone. Try speaking different vowel sounds ("aa", "oo", "ee"). Also try some consonant sounds ("k", "b", "sss"). Sketch (or do screen captures of) the spectrogram patterns for these different sounds, and attach them to your homework. Describe how their spectrogram patterns differ. Also try speaking a couple of words with different sound combinations (e.g., "book", "cars"). Again, sketch or capture pictures of the spectrograms and attach them to the homework. Finally, show where on the spectrograms you can identify the different sounds that make up the words (e.g., "b", "oo", and "k" for "book"). (The features that you use to identify parts of words from spectrograms are thought to be similar to those your brain uses to recognize parts of words from your ears' output signals.)

Notes on using the online spectrogram:

- Try to use the online spectrogram in a quiet room – otherwise, you'll see a lot of interference from ambient sounds.
- A screenshot of the online spectrogram as I was speaking "aa – ee – oo" (with pauses between the vowel sounds) is shown below. A couple of points to notice:
  1. The default settings on spectrograms calculated by MATLAB (the versions shown in the lectures) have frequency on the x-axis and time on the y-axis; the online spectrogram has time on the x-axis and frequency on the y-axis.
  2. Because the online spectrogram is a scrolling real-time version, the rightmost part of the picture represents sounds spoken first (i.e., time becomes later as you move left in the spectrogram). In the screenshot below: "aa" is the part

shown between 4 and 5 sec. on the x-axis, “ee” is the part around 3, and “oo” is the part between 1 and 2.

- When you first use the spectrogram app the site will ask to have access to your computer’s microphone. If you do not want to allow this, you can alternatively record speech and generate spectrograms using MATLAB. Instructions for doing this will be posted on Moodle. (Even if you choose the MATLAB option, you might still find it interesting to look at the discussions and videos on the web site.)


kHz
second
Sensitivity (dB)
Pause