The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 ***Version 3.0***

Date: Oct. 13, 2021 Course: EE 445S Evans

Name: **Straits** **Dire**
Last, First

- **Exam duration**. The exam is scheduled to last 75 minutes.
- **Materials allowed**. You may use books, notes, your laptop/tablet, and a calculator.
- **Disable all networks**. Please disable all network connections on all computer systems. You may not access the Internet or other networks during the exam.
- **Electronics.** Power down phones. No headphones. Mute your computer systems.
- **Fully justify your answers**. When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification. You could reference homework solutions, test solutions, etc.
- **Matlab**. No question on the test requires you to write or interpret Matlab code. If you base an answer on Matlab code, then please provide the code as part of the justification.
- **Put all work on the test**. All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Academic integrity.** By submitting this exam, you affirm that you have not received help directly or indirectly on this test from another human except your instructor, Prof. Evans, and that you did not provide help, directly or indirectly, to another student taking this exam.

	Problem	*Point Value*	*Your score*	*Topic*
Pick Withers	1	24		Sinusoidal Generation
David Knopfler	2	24		FIR Filter Design
Mark Knopfler	3	28		Discrete-Time Audio Effects
John Illsley	4	24		Mystery Systems
	Total	100		

Lecture 1 & 4	Lab 2
HW 0.3 & 0.4	F 2016 Midterm 1.2
Lecture Slides 1-16 and 1-19 to 1-22	
Discrete-Time Periodicity	

Problem 1.1. *Sinusoidal Generation*. 24 points.

You're asked to generate one period of a discrete-time cosine signal $y[n]$:

- The continuous-time frequency is 131 Hz ('C' note on the Western scale in the third octave).
- The sampling rate f_s is 8,000 Hz.

(a) What is the discrete-time frequency in rad/sample of the discrete-time cosine signal? *4 points*.

$x(t) = \cos(2\,\pi\,f_0\,t)$ where $f_0 = 131$ Hz.

$x[n] = x(n\,T_s) = x\left(\frac{n}{f_s}\right) = \cos\left(2\,\pi\,\frac{f_0}{f_s}\,n\right) = \cos(\omega_0\,n)$

where $\omega_0 = 2\,\pi\,\frac{f_0}{f_s} = 2\,\pi\,\frac{131}{8000}$ is the discrete-time frequency in rad/sample.

(b) What is the fundamental period of the discrete-time cosine signal in samples? *4 points*.

Per the Handout on Discrete-Time Periodicity, a sinusoidal signal with discrete-time frequency, where the common factors between integers N and L have been removed,

$$\omega_0 = 2\,\pi\,\frac{f_0}{f_s} = 2\,\pi\,\frac{N}{L}$$

has a discrete-time fundamental period of L samples. In the period of L samples, there are N continuous-time periods of a continuous-time sinusoidal signal at frequency f_0.

(c) Give a difference equation whose impulse response will generate the discrete-time cosine signal. *4 points*. **From homework problem 0.4, $y[n] = (2\cos\omega_0)\, y[n-1] - y[n-2] + x[n] - (\cos\omega_0)\, x[n-1]$ with initial conditions $y[-1]$, $y[-2]$, and $x[-1]$ being zero as necessary conditions for LTI.**

(d) Compare the run-time complexity for the difference equation and the lookup table method. The lookup table would store an entire period of cosine values computed offline. *8 points*.

Method	*Total Memory Needed*	*Multiplications per output sample*	*Reads per output sample*	*Writes per output sample*
Difference equation	**7 words**	**2**	**6 words**	**4 words**
Lookup table	**8001 words**	**0**	**1 word**	**1 word**

The difference equation contains two constants, current input value, previous input value, current output value, and two previous output values. Total memory of 7 words.

The difference equation has two multiplications $(2\cos\omega_0)\, y[n-1]$ and $(\cos\omega_0)\, x[n-1]$ per output sample. To compute $y[n]$, the other six values have to be read once each. Also, we'll need to write the result $y[n]$, and update $y[n-1]$, $y[n-2]$ and $x[n-1]$.

Cosine lookup table stores one period of $L = 8000$ samples. We use $n = 0, 1, \ldots, L-1$ to read the precomputed value from the table for $\cos(\omega_0\,n)$ and write it out as $y[n]$.

(e) How would you use the lookup table for the cosine signal to generate a discrete-time sine signal with the same frequency? *4 points*.

With $\sin(\theta) = \cos(\theta - \pi/2)$, we delay the cosine by ¼ of a period: $n_0 = L/4 = 2000$ samples. To generate $\sin(\omega_0\,n) = \cos(\omega_0\,(n - n_0))$, we start with $n = 0$ which is index $-n_0$ into the cosine lookup table, which is out of bounds. Due to periodicity, index $-n_0$ is the same as $-n_0 + L = 6000$ into the cosine table. We start with index 6000 into the cosine table. As we increment the index and reach the end of the lookup table, we go to the first entry (index 0).

Note: The above answer in part (d) was my initial answer and is plenty for a timed test. Upon further analysis given on the next page, I learned that the determination of the value of n_0 is a bit more complicated.

(d) We would like to choose the delay n_0 for $x[n] = \cos(\omega_0\, n)$ to give $y[n] = \sin(\omega_0\, n)$:

$$y[n] = x[n - n_0] = \cos\left(2\,\pi\,\frac{N}{L}\,(n - n_0)\right) = \cos\left(2\,\pi\,\frac{N}{L}n - 2\,\pi\,\frac{N}{L}n_0\right) = \cos(\omega_0\, n - \theta)$$

We would like the phase shift θ to be $\frac{\pi}{2} + 2\pi k$ where k is an integer:

$$2\,\pi\,\frac{N}{L}n_0 = \frac{\pi}{2} + 2\pi k \;\rightarrow\; n_0 = \left(\frac{1}{4} + k\right)\frac{L}{N} = \left(\frac{1+4k}{4}\right)\frac{L}{N} = \left(\frac{1+4k}{4N}\right)L$$

For example, when $N = 1$ and $k = 0$, we would have $n_0 = \frac{L}{4}$. For $N = 131$ and $L = 8000$:

$$n_0 = \left(\frac{1+4k}{4N}\right)L = \left(\frac{1+4k}{N}\right)\frac{L}{4} = \left(\frac{1+4k}{131}\right)2000$$

We can try different values for integer k to find integer n_0. When $k = 98$, $n_0 = 6000$. We would start indexing into the cosine table at -6000 which is same as an index of 2000 due to periodicity.

***Deeper dive*: We would like to have the term below be an integer, which we'll denote as m:**

$$\left(\frac{1+4k}{131}\right) = m \;\rightarrow\; 1 + 4k = 131m \;\rightarrow\; 1 = 131m - 4k$$

The equation $131m - 4k = 1$ is Bezout's identity which can be solved efficiently by Euclid's algorithm. This is not a topic that would have appeared in the course or its pre-requisites.

***Examples*: We'll use a shorter discrete-time period L = 20 to visualize results. Examples show two different values for $-n_0$ to achieve a phase shift of -π/2 for the same value of L. In example #1, $n_0 \neq \frac{L}{4}$. N and L must be relatively prime. To satisfy the Sampling Theorem, $L > 2N$.**

Let N = 7 and L = 20:

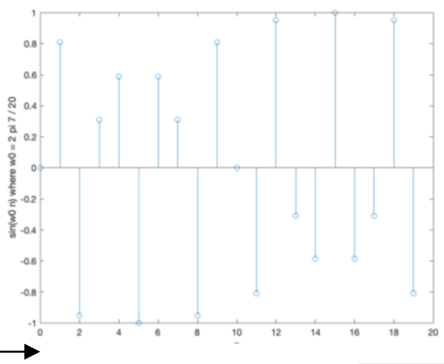
$$\omega_0 = 2\,\pi\left(\frac{7}{20}\right)$$

$$n_0 = \left(\frac{1+4k}{7}\right)5$$

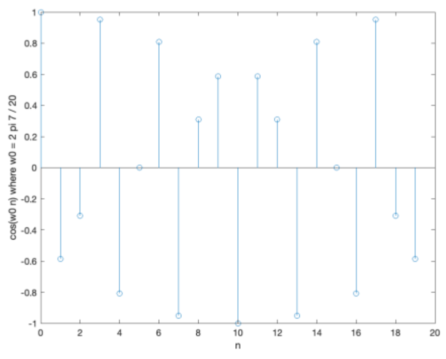
$k = 5$ to give $n_0 = 15$.

Shift by -15 samples is same as shift of 5 samples due to period of 20 samples.

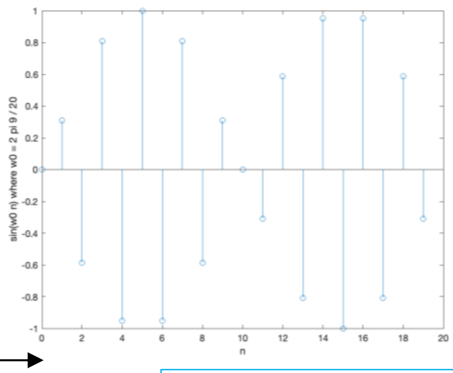
Example #1



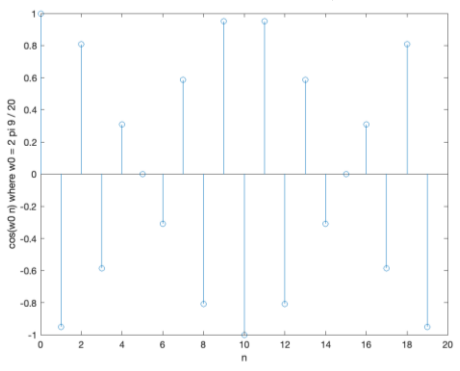
5 samples $\longrightarrow$



Example #2



15 samples $\longrightarrow$



N = 9 and L = 20:

$$\omega_0 = 2\,\pi\left(\frac{9}{20}\right)$$

$$n_0 = \left(\frac{1+4k}{9}\right)5$$

$k = 2$ to give $n_0 = 5$

Shift of -5 samples is same as shift of 15 samples due to period of 20 samples

Lecture Slides 6-5 to 6-10 | In-Lecture #2 Assignment

Labs 2 & 3 | HW 1.1 2.1 2.2 2.3 & 3.1

Problem 1.2 *FIR Filter Design*. 24 points.

You're asked to design a lowpass linear phase finite impulse response (FIR) filter to meet the following specifications:

- Pass frequencies in octave 0, Western scale, from 16.35160 Hz ('C') to 30.8611 Hz ('B')
- Zero out odd harmonics of the 60 Hz powerline frequency (60 Hz, 180 Hz, 300 Hz, etc.)
- Sampling rate is less than 1000 Hz

Midterm Problems: 1.2 Sp 19, 1.3(d) Sp 19, 1.2 F16

(a) What sampling rate would you choose? Why? *12 points*

Lecture Slides 1-16, 1-21 & 3-

Lecture 4

From the Sampling Theorem, sample at $f_s > 2\, f_{max}$ to be able to reconstruct the original signal from its sampled version. Frequencies captured by sampling are $-\frac{1}{2} f_s < f_{max} < \frac{1}{2} f_s$ and these are the frequencies used in reconstructing a signal

Considerations for choosing the sampling rate $f_s < 1000$ Hz :

- **The maximum passband frequency is 30.8611 Hz. $f_s > 2\,(30.8611 \text{ Hz})$**
- **We want to remove odd harmonics of the main powerline frequency of 60 Hz, which are 60 Hz, 180 Hz, 300 Hz, 420 Hz, 540 Hz, 660 Hz, 780 Hz, etc. Some will alias.**
- **Our sampling rate $f_s < 1000$ Hz, so any odd harmonics of 60 Hz above $\frac{1}{2} f_s$ will alias. The harmonics are 60 Hz, 180 Hz, 300 Hz, 420 Hz, 540 Hz, 660 Hz, 780 Hz, 900 Hz, etc.**
- **Use the highest sampling rate possible for better audio quality.**

If we choose a sampling rate between adjacent odd harmonic frequencies, all the odd harmonics that alias will alias to one of the odd harmonic frequencies that didn't alias.

When $f_s = 240$ Hz, the odd harmonic at 60 Hz does not alias. For the other odd harmonics:

- **180 Hz aliases to 180 Hz – 240 Hz = –60 Hz and –180 Hz aliases to –180 Hz + 240 Hz = 60 Hz**
- **300 Hz aliases to 300 – 240 = 60 Hz and –300 Hz aliases –300 + 240 = –60 Hz, etc.**

Choose $f_s = 960$ Hz

(b) Give the coefficients of the FIR filter to meet the specifications. *12 points.*

Lectures 3 5 & 6

Lab #3 | Midterm 1.1 F 2020 Sp 2020 F 2018 F 2016 Sp 2010

HW 0.1 1.1 1.3 2.1 2.2 2.3 & 3.2

JSK Ch. 7

In-Lecture Assignment #2

Designing Averaging Filters

***Solution #1*: Averaging Filter. From the handout Designing Averaging Filters, an averaging filter has a null bandwidth of $f_s\, /\, N$ and we would like the first null to be at 60 Hz so that we can pass octave 0:**

$$\frac{f_s}{N} = 60 \;\rightarrow\; N = \frac{f_s}{60} = \frac{960}{60} = 16$$

The averaging filter would zero out frequencies that are multiples of 60 Hz, which would include the odd and even harmonics. It is zeroing out twice as many frequencies as needed, but still meets the specifications. See next page for additional analysis.

***Solution #2*: Manually place zeros. MATLAB command `poly` can convert a list of roots (zeros) to an unfactored polynomial.**

We place a zero on the unit circle at the angle of each positive and negative discrete-time frequency to be zeroed out. There are four of each. The zeros

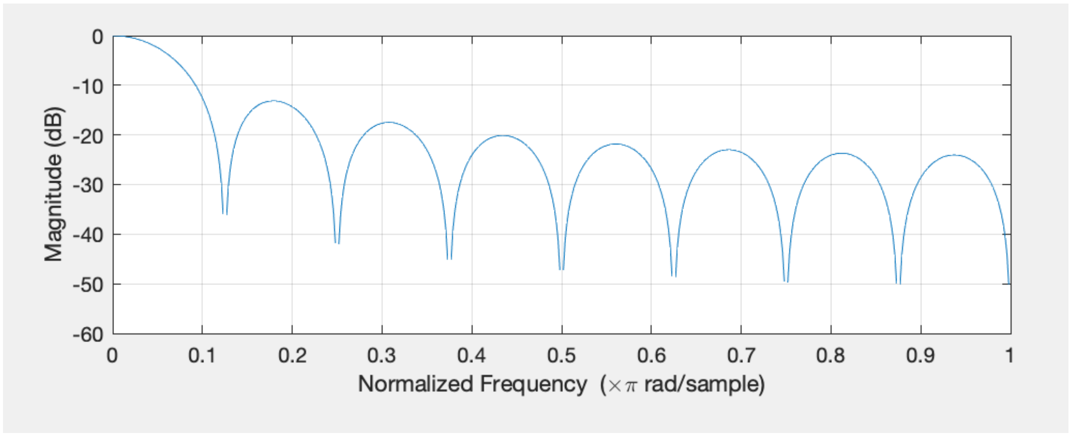
$e^{\pm j\omega_0}, e^{\pm j3\omega_0}, e^{\pm j5\omega_0}, e^{\pm j7\omega_0}$ where $\omega_0 = 2\,\pi\, \frac{60}{960} = \frac{\pi}{8}$ give coefficients [1 0 0 0 0 0 0 0 1].

Filter has linear phase, but multiband in selectivity instead of lowpass. (See next page.)

Additional analysis for problem 1.2(b). This would not be expected on a test.

Solution #1*: *Averaging filter.

```
h = 1/16)*ones(1,16);      % impulse response is a rectangular pulse
freqz(h);                  % frequency response is a periodic sinc
```



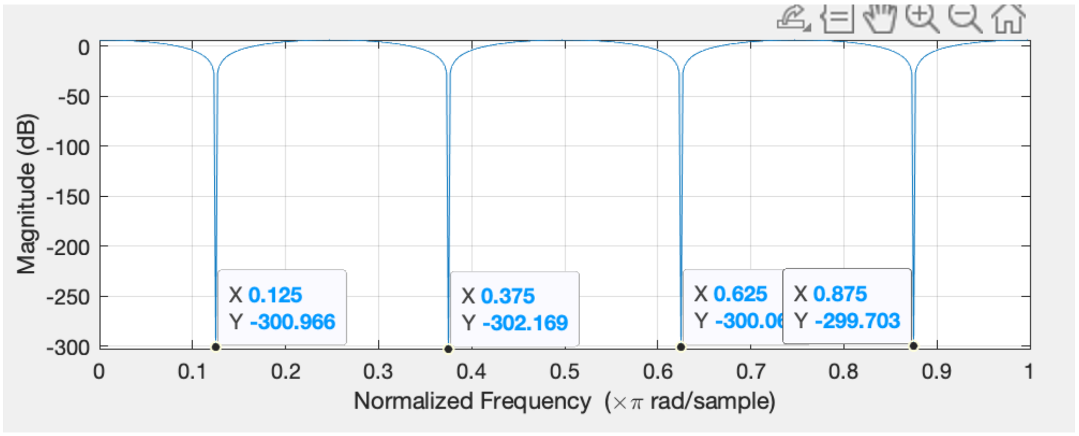
Magnitude response shows discrete-time frequencies at multiples of $\frac{\pi}{8}$ are being zeroed out.

Filter coefficients are even symmetric about the midpoint, which means the FIR filter has linear phase.

Solution #2*: *Manually place zeros to design the FIR filter.

Matlab code to determine the filter coefficients from the zero locations

```
w0 = pi/8;
zerolist = exp( j * w0 * [1 -1 3 -3 5 -5 7 -7] );
h = poly(zerolist)
>> [ 1 0 0 0 0 0 0 0 1 ]
freqz(h);
```



Magnitude response shows discrete-time frequencies $\frac{1}{8}\pi, \frac{3}{8}\pi, \frac{5}{8}\pi$ and $\frac{7}{8}\pi$ being zeroed out.

Filter coefficients [1 0 0 0 0 0 0 0 1] are even symmetric about the midpoint, which means the FIR filter has linear phase.

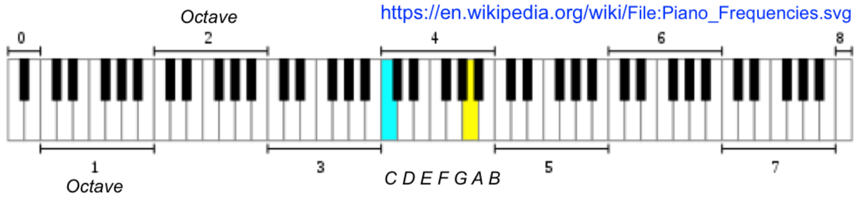
The magnitude response has multiple passbands. It's not lowpass.

This filter is called a <u>comb filter</u>. We'll see it later in lab #7 on guitar effects.

Lecture Slides 6-5 to 6-10	In-Lecture #2 Assignment
Labs 2 & 3	HW 1.1 2.1 2.2 2.3 & 3.1

Problem 1.3 *Discrete-Time Audio Effects.* 28 points.

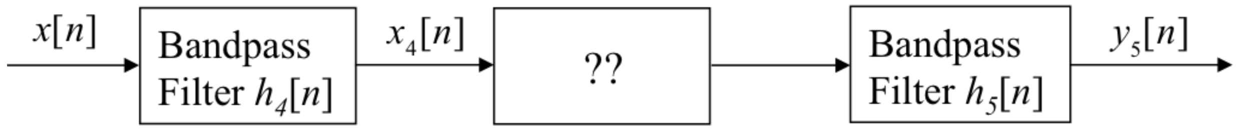
The notes on the Western scale on an 88-key piano keyboard grouped into octaves follow:



The frequency of note A4 (i.e. 'A' in the 4th octave) at 440 Hz is twice the frequency of A3 at 220 Hz.

This type of octave spacing occurs for all the notes on the Western scale.

Design a **discrete-time** audio effects system that will extract the fourth octave of frequencies and then alter that octave of frequencies to be in the next higher octave:



All notes on the fourth octave should appear as the same notes in the next higher octave.

Bandpass filter $h_k[n]$ passes frequencies in the kth octave and attenuates other frequencies.

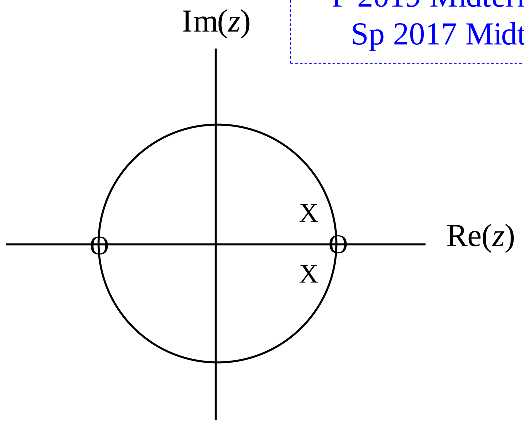
For signals $x[n]$ and $x_4[n]$, the sampling rate f_s is 8,000 Hz.

F 2019 Midterm 1.3(a) Sp 2017 Midterm 1.3

(a) Design a second-order infinite impulse response (IIR) bandpass filter $h_4[n]$ to pass the fourth octave and attenuate the other octaves. In the fourth octave, the lowest note is 262 Hz and highest note is 494 Hz. *12 points.*

i. Give formulas for the pole and zero locations.

ii. Plot poles and zeros on the diagram on the right.



Place a pole p_0 at the center frequency in the fourth octave and near but inside the unit circle to define the passband in positive frequencies:

$$f_{center} = \frac{262\ Hz + 492\ Hz}{2} = 377\ Hz$$

$$\omega_{center} = 2\,\pi\,\frac{f_{center}}{f_s} = 2\,\pi\,\frac{377}{8000}$$

$$p_0 = 0.95\,e^{j\,\omega_{center}} \text{ and } p_1 = 0.95\,e^{-j\,\omega_{center}}$$

Place zeros on the unit circle to define stopbands centered at 0 rad/sample and π rad/sample:

$$z_0 = e^{j\,0} = 1 \text{ and } z_1 = e^{j\,\pi} = -1$$

HW 1.3	Sp 2021 Midterm 1.4(a) Sp 2020 Midterm 1.4(b) F 2016 Midterm 1.3(c) F 2015 Midterm 1.3(a)
	Lecture slides 3-7 & 3-10

(b) What system would you use for the ?? block? Why? *9 points.*

Solution #1: **Squaring block. Effect on single input frequency:**

$$y[n] = x^2[n] = \cos^2(\omega_0\,n) = \frac{1}{2} + \frac{1}{2}\cos(2\,\omega_0\,n)$$

It doubles the input frequency and creates a zero frequency (DC) term.

Any single note in the fourth octave would be doubled to be the same note in the fifth octave, and DC term would be filtered out by the fifth octave bandpass filter.

Solution #2: **Downsampling by 2.**

(c) What would the output be for your proposed system if two notes in the fourth octave were being played at the same time? *7 points*.

Solution #1: **Let the input signal to the squaring block be a sum of two cosine signals at different note frequencies ω_1 and ω_2 in the fourth octave.**

$$x[n] = \cos(\omega_1\, n) + \cos(\omega_2\, n)$$

$$y[n] = x^2[n] = (\cos(\omega_1\, n) + \cos(\omega_2\, n))^2$$

$$y[n] = \cos^2(\omega_1\, n) + 2\cos(\omega_1\, n)\cos(\omega_2\, n) + \cos^2(\omega_2\, n)$$

We know from part (b) that $\cos^2(\omega_0\, n) = \frac{1}{2} + \frac{1}{2}\cos(2\,\omega_0\, n)$.

The middle term $2\cos(\omega_1\, n)\cos(\omega_2\, n)$ is the modulation of one cosine by another. The resulting frequencies are $\omega_1 + \omega_2$ and $|\omega_1 - \omega_2|$.

$$y[n] = 1 + \frac{1}{2}\cos(2\,\omega_1\, n) + \cos\big((\omega_1 + \omega_2)\, n\big) + \cos\big((\omega_1 - \omega_2)\, n\big) + \frac{1}{2}\cos(2\,\omega_2\, n)$$

The output consists of the following frequencies:

$2\,\omega_1$ the frequency for note #1 in the fifth octave
$2\,\omega_2$ the frequency for note #2 in the fifth octave
$\omega_1 + \omega_2$ which will fall in the fifth octave
$|\omega_1 - \omega_2|$ which fall below the fourth octave
0 frequency

The last two frequencies will be attenuated by the bandpass filter for the fifth octave, but there will be intermodulation distortion (or audio effect?) in the fifth octave at frequency $\omega_1 + \omega_2$.

Solution #2: **The two notes in the fourth octave will show up as their respective notes in the fifth octave. Downsampling by 2 will also double the bandwidth around the principal frequency. One can see this is in problem 1.4(b) on this test.**

Please note that this discrete-time audio effect system only alters the principal frequencies of notes. It filters out all the harmonics of the note. It's the harmonics that gives richness and texture to the notes. It's the harmonics that allow us to identify what instrument played it.

Midterm 1: 1.4 Sp11, 1.3 Sp15, 1.5 Sp19, 1.4 F19 & 1.4 Sp20

Lecture 4 | Handout Common Signals in Matlab

HW 1.2 1.3 & 2.2 | In-Lecture #1 Assignment

Problem 1.4. *Mystery Systems.* 24 points.

You're trying to identify unknown discrete-time systems.

You input a discrete-time chirp signal $x[n]$ and look at the output to figure out what the system is.

The discrete-time chirp is formed by sampling a chirp signal that sweeps 0 to 4000 Hz over 0 to 5s

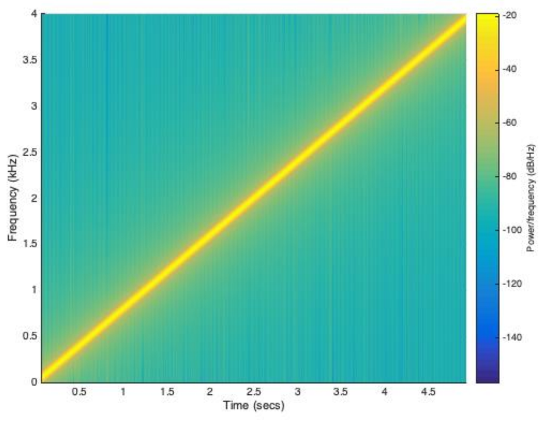
$$x(t) = \cos(2\pi f_1 t + 2\pi \mu t^2)$$

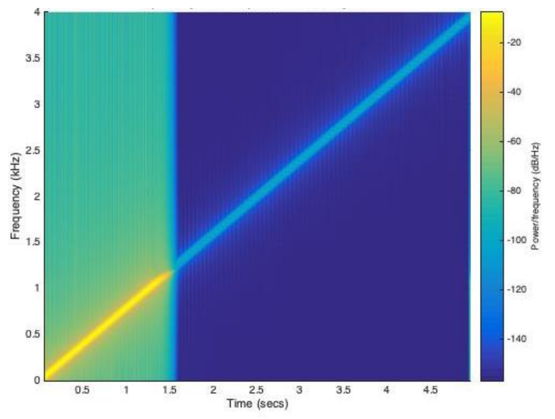
where $f_1 = 0$ Hz, $f_2 = 4000$ Hz, and $\mu = \frac{f_2 - f_1}{2\, t_{\max}} = \frac{4000 \text{ Hz}}{10\ s} = 400 \text{ Hz}^2$. Sampling rate f_s is 8000 Hz.

In each part below, identify the unknown system as one of the following:

1. filter – give selectivity (lowpass, highpass, bandpass, bandstop) and passband/stopband frequencies
2. upsampler – give upsampling factor
3. downsampler – give downsampling factor

(a) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). *12 points.*





Lectures 1 3 5 & 6

Lab #3

HW 0.1 1.1 1.3 2.1 & 2.2

JSK Ch. 7

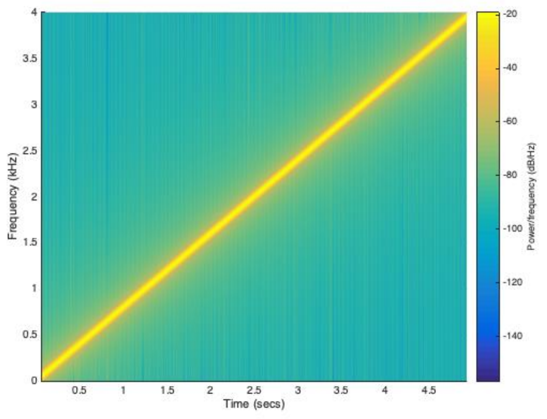
Midterm 1.1 F 2020 Sp 2020 F 2018

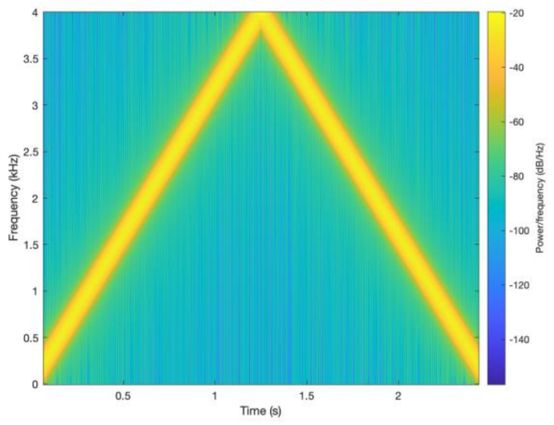
Designing Averaging Filters

From the output spectrogram, frequencies from 0 Hz to about 1200 Hz are passed. Principal frequencies in the chirp above about 1300 Hz are severely attenuated and their medium blue color indicates at attenuation vs. bright yellow of about 100 dB.

Lowpass Filter with passband frequency ~1200 Hz and stopband frequency ~1300 Hz.

(b) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). *12 points.*





Lecture

HW 0.3 & 2.2

Midterm Problems 1.2 F 18 1.2(d) Sp 18 1.2(d) F 09

When compared to the input spectrogram, the output spectrogram has half the duration in time and its principal frequency is increasing. From 1.25s to 2.5s, aliasing occurs.

Downsampling by 2, per homework problem 2.2(d).

Matlab code to generate the spectrograms for problem 1.4.

(a) Lowpass filter

```
fs = 8000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% Design lowpass filter
fnyquist = fs/2;
fpass = 1000;
fstop = 1200;
ctfrequencies = [0 fpass fstop fnyquist];
idealAmplitudes = [1 1 0 0];
pmfrequencies = ctfrequencies / fnyquist;
filterOrder = 200;
h = firpm( filterOrder, pmfrequencies, idealAmplitudes );
h = h / sum(h .^ 2);

y = conv(x, h);

%%% Plot spectrogram of signal
blockSize = 1024;
overlap = 1023;
figure;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
```

(b) Downsampling by 2

```
fs = 8000;
Ts = 1 / fs;
tmax = 5;
t = 0 : Ts : tmax;

%% Create chirp signal
f1 = 0;
f2 = fs/2;
mu = (f2 - f1) / (2*tmax);
x = cos(2*pi*f1*t + 2*pi*mu*(t.^2));

%% Downsampling by 2
y = x(1:2:end);

blockSize = 1024;
overlap = 1023;
spectrogram(y, blockSize, overlap, blockSize, fs, 'yaxis');
```