The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1 ***Solutions Version 2.0***

Date: March 9, 2022 Course: EE 445S Evans

Name: **Matrix**, **The**
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*
Morpheus	1	24		Sinusoidal Generation
Neo	2	26		Filter Analysis
Trinity	3	26		Chromagram Filter Design
Agent Smith	4	24		Mystery Systems
	Total	100		

Fall 2021 Midterm 1.1 & Fall 2016 Midterm 1.2	Lectures 1 & 4	Lab 2
HW 0.3, 0.4, 1.1(d) & 2.1(d)	Lecture Slides 1-16 and 1-19 to 1-22	
Discrete-Time Periodicity Handout		

Problem 1.1. *Sinusoidal Generation*. 24 points.

You're asked to generate one period of a discrete-time sine signal $y[n] = \sin(\omega_0 n)$:

- The continuous-time frequency is 196 Hz ('G' note on the Western scale in the third octave).
- The sampling rate f_s is 8000 Hz.

(a) What is the discrete-time frequency ω_0 in rad/sample of the discrete-time sine signal? *4 points*.

$\boldsymbol{y(t) = \sin(2\,\pi\, f_0\, t)}$ **where** $\boldsymbol{f_0 = 196}$ **Hz.**

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

where $\boldsymbol{\omega_0 = 2\,\pi\,\frac{f_0}{f_s} = 2\,\pi\,\frac{196}{8000}}$ **is the discrete-time frequency in rad/sample.**

(b) What is the fundamental period of the discrete-time sine 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,

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

has a discrete-time fundamental period of *L* samples:

$$\boldsymbol{\omega_0 = 2\,\pi\,\frac{f_0}{f_s} = 2\,\pi\,\frac{196\text{ Hz}}{8000\text{ Hz}} = 2\,\pi\,\frac{49}{2000}}$$

In the discrete-time period of *L* = 2000 samples, there are *N* = 49 continuous-time periods of a continuous-time sinusoidal signal at frequency $\boldsymbol{f_0}$**.**

(c) Give a difference equation whose impulse response will generate the discrete-time sine signal. *4 points*. **From lab #2, $y[n] = (2\cos\omega_0)\, y[n-1] - y[n-2] + (\sin\omega_0)\, x[n-1]$ for $n \geq 0$ with initial conditions $y[-1]$, $y[-2]$, and $x[-1]$ being zero as necessary conditions for LTI to hold. This difference equation comes from** $\boldsymbol{Z\{\sin(\omega_0 n)\, u[n]\} = \frac{\sin(\omega_0)\, z^{-1}}{1 - 2\cos(\omega_0)\, z^{-1} + z^{-2}}}$ **for** $\boldsymbol{|z| > 1}$**.**

(d) An alternate method to compute the amplitude value is to use the Taylor series expansion for the sine function

$$\sin(\theta) = \theta - \frac{1}{3!}\theta^3 + \frac{1}{5!}\theta^5 - \frac{1}{7!}\theta^7 + \frac{1}{9!}\theta^9 - \cdots$$

and keep a finite number of terms. For good quality over one period of θ, we'll need 17 terms, i.e. from the θ term to the θ^{33} term.

Compare the run-time complexity for the difference equation and the lookup table method. The lookup table would store an entire period of sine values computed offline. *12 points*.

Method	Total Memory Needed	Multiplications per output sample	Reads per output sample	Writes per output sample
Taylor series	**18**	**291**	**19**	**1**
Difference equation	**7**	**2**	**6**	**4**
Lookup table	**2000**	**0**	**1**	**1**

Taylor series method would compute $\boldsymbol{\theta = \omega_0\, n}$ **using 1 multiplication, and perform a modulo operation (1 multiplication by** $\boldsymbol{1/(2\pi)}$ **and 1 additional multiplication) to get the value of angle** $\boldsymbol{\theta}$ **in the first period of** $\boldsymbol{[0, 2\pi)}$**. For a Taylor series expansion with *N* non-zero terms,**

$$\boldsymbol{\sin(\theta) = \theta - \frac{1}{3!}\theta^3 + \frac{1}{5!}\theta^5 - \frac{1}{7!}\theta^7 + \frac{1}{9!}\theta^9 - \cdots}$$

we'll assume the constants $1/m!$ have been computed offline. When $N = 1$, $\sin(\theta) = \theta$ does not need any multiplications; $N = 2$ terms needs 3 mults; $N = 3$ needs 3+5=8 mults; $N = 5$ needs 3+5+7=15 mults; $N = 7$ needs 3+5+7+9=24 mults; or $(N+1)(N-1)$ mults in general.

<u>Difference equation</u> contains two constants, previous input value, current output value, two previous output values. The current input is stored into the previous input value. Total memory of 7 words. The difference equation has two multiplications ($2 \cos \omega_0$) $y[n-1]$ and ($\sin \omega_0$) $x[n-1]$ 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 $x[n-1]$.

<u>Lookup table</u> stores one period of $L = 2000$ samples. We use $n = 0, 1, \ldots, L-1$ to read the precomputed value from the table for $\sin(\omega_0\, n)$ and write it out as $y[n]$.

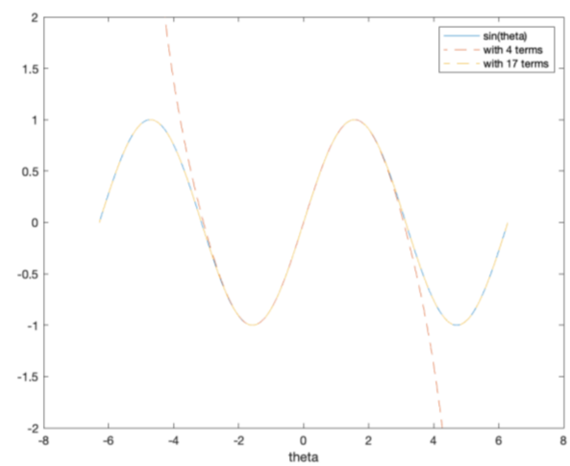
Epilogue for part (d): Although not asked, we can reduce the number of multiplications by using Horner's form of the series expansion which results from iteratively factoring out θ^2:

$$\sin(\theta) = \theta - \theta^2 \left(\frac{1}{3!}\theta + \frac{1}{5!}\theta^3 - \frac{1}{7!}\theta^5 + \frac{1}{9!}\theta^7 - \cdots \right)$$

$$\sin(\theta) = \theta - \theta^2 \left(\frac{1}{3!}\theta + \theta^2 \left(\frac{1}{5!}\theta - \frac{1}{7!}\theta^3 + \frac{1}{9!}\theta^5 - \cdots \right) \right)$$

In this case, Horner's form would have a nested structure of $a_n \theta + b_n \theta^2$ for $n \in [2, N]$. The $N-1$ nested terms would need $2(N-1)$ multiplications. We need one multiplication to compute θ^2, for a total of $2(N-1)+1$ multiplications. For $N = 17$ terms, 33 multiplications would be needed for the polynomial calculation, plus 3 multiplications for the calculation of $\theta = \omega_0\, n$ and the modulo operation, for a total of 36 multiplications.

Taylor series expansion is about the origin, i.e. $\theta = 0$. For the Taylor series to provide a good fit to $\sin(\theta)$ for $\theta \in [0, 2\pi]$, 17 terms are needed; however, to provide a good fit for $\theta \in [-\pi, \pi]$, only 4 terms are needed (from the θ to the θ^7 terms). This makes sense because the Taylor series expansion is about $\theta = 0$. See <u>https://en.wikipedia.org/wiki/Taylor_series</u> and below.




```
theta = -2*pi : (4*pi)/1000 : 2*pi;
maxi = 17;
sinapprox = zeros(maxi, length(theta));
sinapprox(1,:) = theta;
for i = 2 : maxi
    term = theta.^(2*i-1) / factorial(2*i-1);
    sinapprox(i,:) = sinapprox(i-1,:) + (-1)^(i+1) * term;
end
plot(theta, sin(theta), '-', ...
     theta, sinapprox(4,:), '--', ...
     theta, sinapprox(maxi,:), '--');
xlabel('theta');
ylim([-2, 2]);
legend('sin(theta)', 'with 4 terms', 'with 17 terms' );
```

In Matlab, the pattern of three dots ... means that the command continues on the next line.

Problem 1.2 *Filter Analysis.* 26 points.

In an electric oven, transfer of energy from a heating element to food is governed by the heat equation.

However, an approximate model is given by the following causal linear time-invariant discrete-time filter with input $x[n]$ and output $y[n]$

$$y[n] = \sum_{m=0}^{n-1} \frac{x[m]}{\alpha^{n-m}} \quad \text{for } n \geq 0$$

where $x[n]$ represents the power delivered to the heating element, α is the thermal diffusivity, and $y[n]$ represents the temperature of the food where a temperature of 0 means room temperature.

We expand the summation as follows

$$y[n] = \frac{1}{\alpha} x[n-1] + \frac{1}{\alpha^2} x[n-2] + \cdots \quad \text{for } n \geq 0$$

and convert $y[n]$ to a recursive difference equation

$$y[n] = \frac{1}{\alpha} x[n-1] + \frac{1}{\alpha} y[n-1] \quad \text{for } n \geq 0$$

For this problem, assume that $\alpha = 2$. We observe the system starting at $n = 1$.

(a) Assume that the food is initially at room temperature. What are the initial conditions and their values? Why? *4 points*. **We observe system starting at $n = 1$: $y[1] = \frac{1}{2} x[0] + \frac{1}{2} y[0]$. Initial conditions $x[0]$ and $y[0]$ should be 0 as necessary conditions for linear and time-invariance (LTI) to hold.**

(b) Give a formula for the impulse response of the filter $h[n]$. Simplify any summations. *6 points*.

To find the impulse response, input an impulse signal, i.e. let $x[n] = \delta[n]$:

$$h[n] = \frac{1}{2} \delta[n-1] + \frac{1}{2} h[n-1] \ \text{ for } n \geq 1$$

where the initial condition $h[0] = 0$. We can compute this iteratively for $n \geq 1$

$$h[1] = \frac{1}{2}\delta[0] + \frac{1}{2} h[0] = \frac{1}{2} \text{ and } h[2] = \frac{1}{2}\delta[1] + \frac{1}{2} h[1] = \frac{1}{2^2} \text{ etc.}$$

$$h[n] = \frac{1}{2^n} u[n-1] = \left(\frac{1}{2}\right)^n u[n-1]$$

(c) Is this a finite impulse response (FIR) or infinite impulse response (IIR) filter? *4 points*.

IIR due to feedback. That is, the output $y[n]$ depends on the previous output value $y[n-1]$.

(d) Derive a formula for the transfer function in the z-domain and the region of convergence. *6 points*.

***Solution #1*: Take the *z*-transform of the impulse response.**

$$H(z) = Z\left\{\left(\frac{1}{2}\right)^n u[n-1]\right\} = \frac{1}{2} Z\left\{\left(\frac{1}{2}\right)^{n-1} u[n-1]\right\} = \frac{1}{2} z^{-1} Z\left\{\left(\frac{1}{2}\right)^n u[n]\right\} = \frac{\frac{1}{2} z^{-1}}{1 - \frac{1}{2} z^{-1}} \text{ for } |z| > \frac{1}{2}$$

***Solution #2*: Take the *z*-transform of the difference equation.**

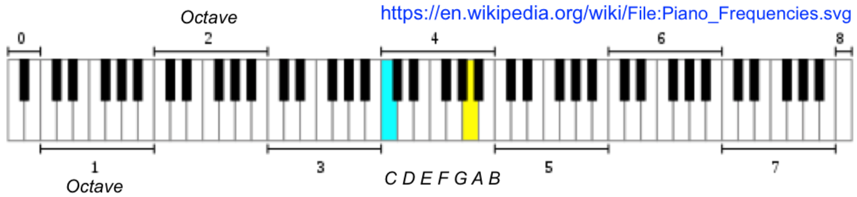
$$Y(z) = \frac{1}{2} z^{-1} X(z) + \frac{1}{2} z^{-1} Y(z) \text{ which gives } H(z) = \frac{Y(z)}{X(z)} = \frac{\frac{1}{2} z^{-1}}{1 - \frac{1}{2} z^{-1}} \text{ for } |z| > \frac{1}{2}$$

(e) Give the frequency selectivity of filter (lowpass, highpass, bandpass, bandstop, allpass, notch) and explain your reasoning. *6 points*. **The transfer function has a pole at z = ½ and a zero at z = 0. Pole angle of 0 rad/sample gives center of passband. This corresponds to a lowpass filter.**

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 *Chromagram Filter Design.* 26 points.

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



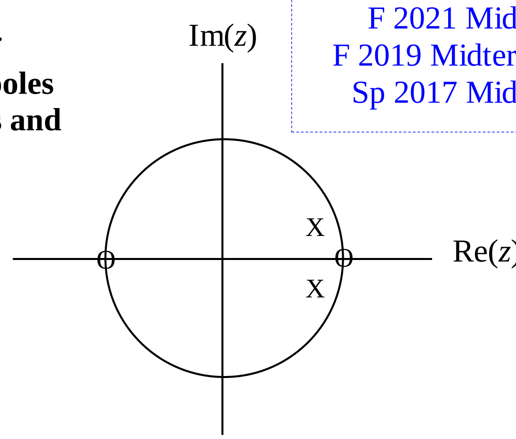
The frequency of note A3 (i.e. 'A' in the 3rd octave) at 220 Hz is twice the frequency of A2 at 110 Hz.

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

For this problem, assume that the sampling rate f_s is 16,000 Hz.

F 2021 Midterm 1.3
F 2019 Midterm 1.3(a)
Sp 2017 Midterm 1.3

When poles and zeros are separated in angle, the angles of poles near but inside the unit circle indicate passband frequencies and the angles of the zeros on or near the unit circle indicate the stopband frequencies, per lecture 6 slides 6-8 to 6-10.



(a) Design a second-order infinite impulse response (IIR) to extract the note A5 (880.0 Hz). The filter should suppress all other frequencies including the neighboring notes G#5 (830.6 Hz) and A#5 (932.3 Hz). *12 points.*
   i. Give formulas for the pole and zero locations.
   ii. Plot poles and zeros on the diagram on the right.

Two zeros at $z = \pm 1$ and two poles at $z = 0.95\, e^{\pm j\,\omega_{A5}}$ where $\omega_{A5} = 2\pi \times \frac{880}{16000}$

(b) To extract the note A4, how would the design of the filter in part (a) change? *6 points.*

Use $\omega_{A4} = 2\pi \times \frac{440}{16000}$ for placement of the poles, which is half the angles of the poles in (a). Poles will move toward the real axis but at the same magnitude as the poles in (a).

Lecture Slides 5-21 to 5-25

(c) Design a first-order discrete-time IIR filter to perform the following smoothing operation

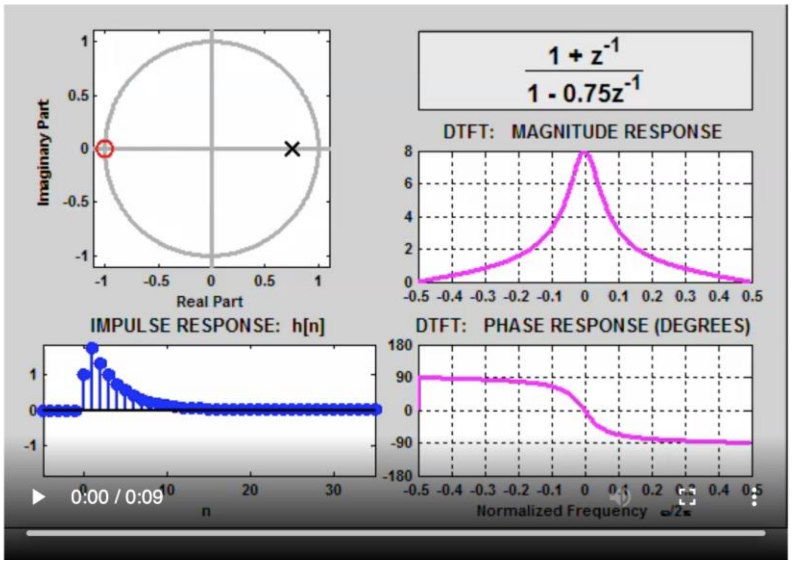
$$y_A[n] = \text{Smooth}\{v[n]\} \text{ where } v[n] = \sum_{k=1}^{6} \left(x[n] * h_{A_k}[n]\right)^2$$

where $h_{A_k}[n]$ is the impulse response of an IIR filter that extracts the note Ak from audio signal $x[n]$, and $y_A[n]$ is the output of the smoothing filter for A notes A1, A2, … A6. This operation is used to construct a chromagram to analyze musical recordings. *8 points.*

A smoothing filter is a lowpass filter; that is, it smooths out sudden changes in the data which corresponding to high-frequency information. In class, we have seen examples of applying an averaging filter, which is a lowpass filter, to create a seven-day moving average of new COVID-19 cases in the Designing Averaging Filters handout and smooth/blur an image in the "Cascading Two FIR Filters" *DSP First* demo mentioned in lecture 5 slide 5-21.

For a first-order IIR filter, place the pole at $z = 0.9$ for a passband centered at 0 rad/sample and the zero at $z = -1$ for a stopband centered at π rad/sample.

We had seen a similar lowpass first-order IIR filter in a DSP First demo (more on next page).

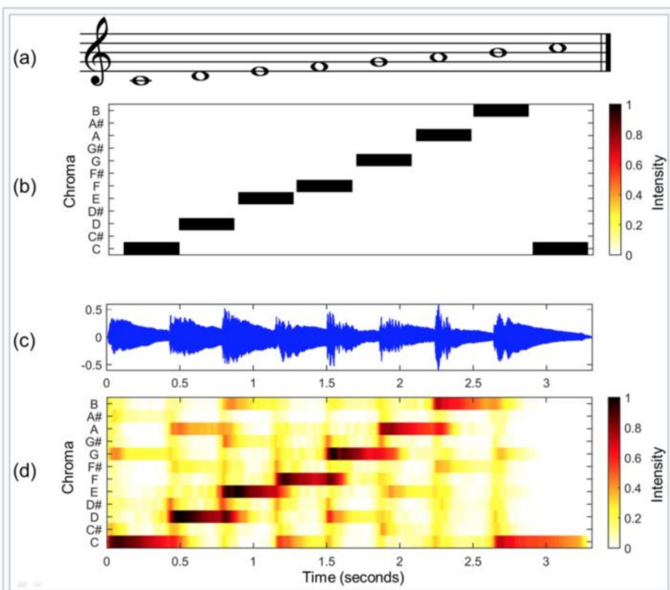


DSP First "[Three-Domain Connections]" demo "IIR filter with one pole and one zero." In this demo, the pole is at $z = 0.75$ and the zero is at $z = -1$ for a lowpass response. See [lecture 6 slide] 6-10.

Epilogue: From Wikipedia ["Chroma Feature"] (Chromagram):

> "In Western music, the term ***chroma feature*** or ***chromagram*** closely relates to the twelve different pitch classes. Chroma-based features, which are also referred to as "pitch class profiles", are a powerful tool for analyzing music whose pitches can be meaningfully categorized (often into twelve categories) and whose tuning approximates to the equal-tempered scale. One main property of chroma features is that they capture harmonic and melodic characteristics of music, while being robust to changes in timbre and instrumentation."

The chromagram (as shown on the right) is a tool used to analyze musical recordings based on the equal temperament scale. It is similar to the spectrogram except that it has exactly twelve frequency bins (corresponding to twelve notes on the Western scale). For each note, all harmonics are combined into the same frequency bin.



(a) Musical score of a C-major scale. (b) Chromagram obtained from the score. (c) Audio recording of the C-major scale played on a piano. (d) Chromagram obtained from the audio recording.

Midterm 1: 1.4 Sp11, 1.3 Sp15, 1.5 Sp19, 1.4 F19, 1.4 Sp20, 1.4 F21

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 **with justification**:

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

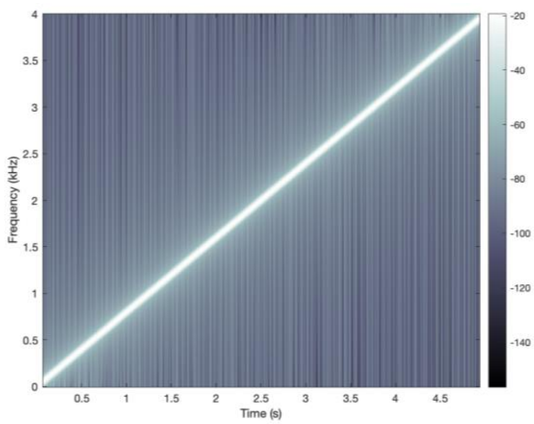
Lab #3

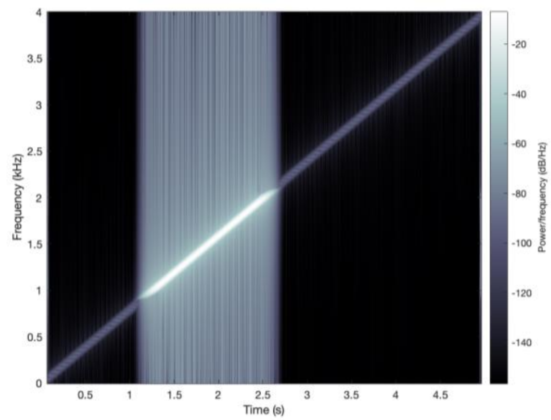
HW 0.1 1.1 1.3 2.1 & 2.2

JSK Ch. 7

Midterm 1.1 F 2020 Sp 2020 F 2018

Designing Averaging Filters





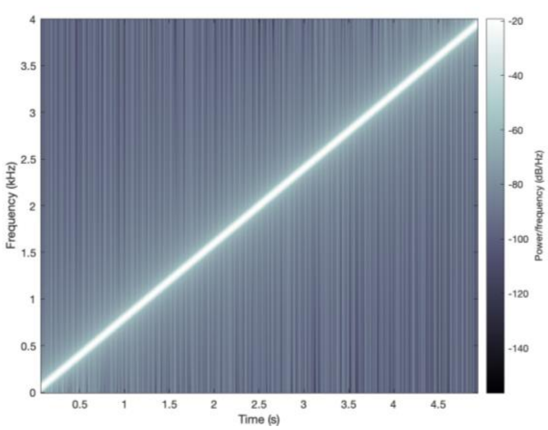
From the output spectrogram, frequencies from about 900 to 2100 Hz are passed. Principal frequencies below 900 Hz and above 2100 Hz are severely attenuated. In the grayscale color map, white has highest magnitude value. This is a bandpass filter.

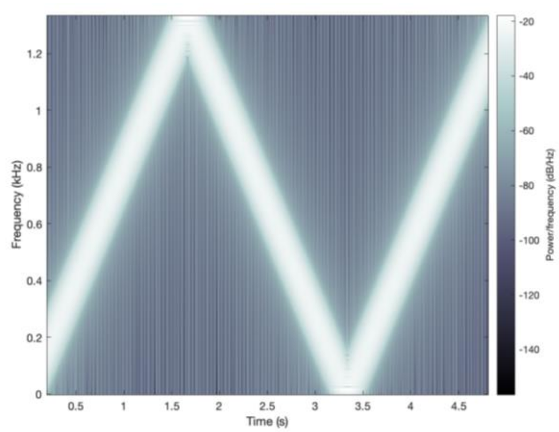
(b) Given spectrograms of the chirp input signal $x[n]$ (left) and output signal $y[n]$ (right). *12 points.*

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 about one-third the range of frequencies values and the principal frequency is a chirp pattern that is wider and that has aliasing. Downsampling by 3 per homework problem 2.2(d).

Epilogue: Matlab code to generate the spectrograms for problem 1.4.

(a) Bandpass 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;
fstop1 =  900;
fpass1 = 1100;
fpass2 = 1900;
fstop2 = 2100;
ctfrequencies = [0 fstop1 fpass1 fpass2 fstop2 fnyquist];
idealAmplitudes = [0 0 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 3

```
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 3
y = x(1:3:end);
blockSize = 1024;
overlap = 1023;
spectrogram(y, blockSize, overlap, blockSize, fs/3, 'yaxis');
```