

Homework #1 Sinusoids, Transforms and Transfer Functions

Assigned Friday, September 13, 2024, and due Friday, September 20, 2024, by 11:59pm

Late homework will be subject to a penalty of 2 points per minute late.

Through midterm #1, here are key sections from JSK's *Software Receiver Design*, and signals & systems textbooks Oppenheim & Willsky's *Signals & Systems* (2nd ed); McClellan, Schafer & Yoder's *Signal Processing First* and Lathi & Green's *Linear Systems & Signals* (3rd ed). **Bold are topics for this homework.**

Topic	<u>Lect.</u>	JSK	O&W	SP First	L&G	<u>Handouts</u>
Introduction	<u>0</u>	Ch. 1				
Bandwidth	<u>1</u>	2.2	4.3-4.4	11-4 to 11-8	6.3-1, 7.2, 7.3, 7.9	Fourier Transforms: Intro , Dictionary and w vs. f
Sinusoidal generation	<u>1</u>	3.2				
Upconversion and downconversion	<u>1&4</u>	2.3-2.6, 3.6; Ch. 5; 6.1-6.4	8.1-8.4	11-8.2	7.3, 7.7	Sinusoidal Ampl. Modulation Example & Summary
Communication sys.	<u>1</u>	Ch. 1-2	Ch. 8		7.7	
Basic CT signals	<u>3</u>	2.10, 4.3	1.3-1.4	2-3, 2-5, 4-4 & 9-1	1.4	Common Signals in Matlab
CT system properties	<u>3</u>		1.6	5-5	1.7	LTI System Properties
Basic DT signals	<u>3</u>		1.3-1.4	4-2.1 & 5-3.2	3.3	Common Signals in Matlab ; DT Periodicity ; Chirp Signals
DT system properties	<u>3</u>		1.6	9-4	3.4-1	Time-Invariance ; LTI System Properties
Fundamental Theorem of Linear Systems	<u>3&5</u>	3.5	3.2	6-1 & 10-1	2.4-4 & 3.8-2	Fundamental Theorem of Linear Systems
Sampling theorem	<u>4</u>		7.1	4-1 4-2 4-5	8.1	
Sampling and aliasing	<u>4</u>	2.8, 3.4, 6.1	7.3 & 7.4	12-3	8.2	Sampling Unit Step Signal
Bandpass sampling	<u>4</u>					
DT-to-CT conversion	<u>4</u>	2.10 & 6.4	7.2	4-4	8.2	
CT convolution	<u>5</u>	4.4 & 4.5	2.2	9-6 & 9-7	2.4	Convolution Example
DT convolution	<u>5</u>	4.4 & 4.5	2.1	5-3.3 & 5-6	3.8	Convolution Example and Four ways to filter a signal
Z-transforms	<u>5&6</u>	App. A.4 & F	10.1-10.3, 10.5	7-1 to 7-5	5.1-5.2	
Transfer functions	<u>5&6</u>	4.5	10.7 & 10.8	8-3, 8-4 & 8-9	5.3	Designing Averaging Filters ; LTI Systems & Freq. Resp.
Relationship between z & Fourier transform	<u>5</u>	App. F.2	10.4	7-6, 8-5, 8-6, 8-10	5.5	
DT FIR filter design and implementation	<u>5</u>	2.12, 3.3, 4.2	10-9	7-7	5.4	Four ways to filter a signal ; Designing Averaging Filters
DT FIR filter analysis	<u>5</u>	Ch. 7 & App.G		7-7 to 7-9		Designing Averaging Filters
DT stability of LTI systems	<u>5&6</u>		10.7.2	8-2.4, 8-4.2, 8-8	3.9 & 3.10	Bounded-Input Bounded-Output Stability
DT IIR filter design by pole-zero placement	<u>6</u>		10.4	8-9 & 8-10	5.6	All-pass Filters
DT classical IIR filter design methods	<u>6</u>				5.10	Elliptic IIR filter design
IIR filter implement.	<u>6</u>		10-9	8-9	5.4	Realizations of IIR Filters

Please read the [homework hints](#)

Office hours for the teaching assistants and Prof. Evans; **bold** indicates a 30-minute timeslot. Prof. Evans' weekly coffee hours are on Fridays 12-2pm.

<i>Time Slot</i>	<i>Monday</i>	<i>Tuesday</i>	<i>Wednesday</i>	<i>Thursday</i>	<i>Friday</i>
10:30am	Evans (ECJ 1.318)		Evans (ECJ 1.318)		
11:00 am	Evans (ECJ 1.318)		Evans (ECJ 1.318)		
12:00 pm					Evans coffee hours (EER Cafe)
1:00 pm			Evans (EER 6.882 & Zoom)		Evans coffee hours (EER Cafe)
2:00 pm			Evans (EER 6.882 & Zoom)	Eun (TBA)	
3:00 pm			Barati (EER 1.810)	Eun (TBA)	
3:30 pm			Barati (EER 1.810)	Barati (TBA)	
4:00 pm			Barati (EER 1.810)	Barati (TBA)	Eun (EER 1.810)
4:30 pm				Barati (TBA)	Eun (EER 1.810)
5:00 pm					Eun (EER 1.810)
5:30 pm					

In your solutions, please put all work for problem 1 together, all work for problem 2 together, etc.

Please submit any MATLAB code that you have written for the homework solution. In your solutions, please put all work for problem 1 together, then all work for problem 2 together, etc.

As stated on the course descriptor, “Discussion of homework questions is encouraged. Please be sure to submit your own independent homework solution.”

1. Transfer Functions. 48 points.

With $x[n]$ denoting the input signal and $y[n]$ denoting the output signal, give the difference equation relating the input signal to the output signal in the discrete-time domain, give the initial conditions and their values, and find the transfer function in the z -domain and the associated region of convergence for the z -transform function, for the following linear time-invariant discrete-time systems:

- (a) Causal averaging filter with five coefficients. See [Designing Averaging Filter Handout](#). 12 points.
- (b) Causal discrete-time approximation to a first-order differentiator. See lecture slide 3-23. 12 points.
- (c) Causal discrete-time approximation to first-order integrator. See [homework hints](#). 12 points.
- (d) Causal bandpass filter with center frequency ω_0 given by the input-output relationship

$$y[n] = (2 \cos \omega_0) r y[n-1] - r^2 y[n-2] + x[n] - (\cos \omega_0) x[n-1]$$

where $0 < r < 1$. Here, r is the radius of the two pole locations. 12 points.

An LTI system is uniquely defined by its impulse response $h[n]$ whose z -transform is $H(z)$. Transfer functions of the form $H(z) = Y(z) / X(z)$ only apply for linear time-invariant (LTI) systems.

Please read the [homework hints](#)

Comment: The linear time-invariant (LTI) system in (d) whose input-output relationship is

$$y[n] = (2 \cos \omega_0) r y[n-1] - r^2 y[n-2] + x[n] - (\cos \omega_0) x[n-1]$$

has several applications. When $r = 1$, the impulse response of the LTI system is $\cos(\omega_0 n) u[n]$.

Hence, the LTI system can be used as a sinusoidal generator. For $r = 1$, the system is not bounded-input bounded-output (BIBO) stable. If $\cos(\omega_0 n) u[n]$ were the input signal, resonance would lead to unbounded amplitude on the output. (Resonance does not always lead to an unbounded output.)

The unbounded response to input $\cos(\omega_0 n) u[n]$ can be used to our advantage. If the filter output were to grow very large in absolute value, then we know that the input signal would have a component equal or at least approximately equal to $\cos(\omega_0 n) u[n]$. The BIBO instability would allow us to detect a sinusoid. Applications of detecting sinusoidal tones in a signal include identification of notes in music, tracking of frequency hopping (e.g. in Bluetooth) and touchtone telephone signal decoding. In practice, we use $r \approx 1$ (e.g. $r = 0.95$) to have good frequency selectivity (i.e. a narrow passband).

2. Chirp Signals. 20 points.

Before starting this problem, please read slides 1-14 to 1-20 on chirp signals and spectrograms (which also contain a lot of examples of Matlab code) in the [Common Signals in Matlab](#) slide deck.

A chirp signal is used in sonar, radar, indoor positioning, LTE, and test/measurement systems. A chirp signal is a sinusoid whose principal frequency increases (or decreases) over time:

$$c(t) = \cos(\theta(t)) \quad \text{where } \theta(t) = 2\pi(f_0 + \frac{1}{2}f_{\text{step}}t)t = 2\pi f_0 t + \pi f_{\text{step}} t^2$$

The principal frequency is f_0 when $t = 0$ and then changes over time at a rate of f_{step} in units of Hz/s. The principal frequency of a sinusoid at a given point in time is called the instantaneous frequency, and it is defined as $d\theta(t) / dt$ in units of rad/s. Here, $d\theta(t) / dt = 2\pi f_0 + 2\pi f_{\text{step}} t = 2\pi(f_0 + f_{\text{step}} t)$.

- Generate a chirp signal for 10s with $f_0 = 20$ Hz and $f_{\text{step}} = 420$ Hz/s. Use sampling rate of 44100 Hz. It will sweep frequencies on an 88-key piano: https://en.wikipedia.org/wiki/Piano_key_frequencies
- Plot the chirp signal in time and frequency domains using `plotspec` from the JSK book.
- Play the chirp signal as an audio signal. You might consider using headphones so as not to bother folks around you. Provide the Matlab code. Describe what you hear.
- Plot the chirp signal in the time-frequency domain using the `spectrogram` function in Matlab and describe the visual representation.

Each part is worth 5 points. This problem is inspired by JSK, Exercise 3.8 on page 46.

3. Spectral Analysis for a Squaring Block. 31 points.

A squaring block can be used for downconversion of an upconverted signal to a baseband signal. In that case, the squaring block would be followed by a lowpass filter, as mentioned on lecture slide 3-7.

JSK Section 3.5 (page 52) analyzes a squaring block with input of a two-sided sinusoid at constant frequency. At the output, half of the input power appears at DC (zero frequency) and the other half appears at twice the input frequency. The output has a DC offset. If the input had two frequencies f_1 and f_2 , the output will have frequencies 0, $|f_1 - f_2|$, $2f_1$, $f_1 + f_2$, and $2f_2$. Again, there is a DC offset. The DC offset can be an issue when the output is an integer, or when floating-point values are converted to an integer, because the DC offset would take up many of the bits in the integer representation, thereby leaving very few bits to represent the rest of the signal. The DC offset can also be an issue during playback. For example, the Matlab `sound` command clips amplitudes outside the range $[-1, 1]$.

This problem uses the gong waveform that accompanies the JSK book as the message signal, and uses the [upconvertGong.m](#) file for upconversion and downconversion that I had demonstrated in class. In order for the squaring approach to work for downconversion, a DC offset has to be added to the message signal in the transmitter. (This is how AM radio works.) In [upconvertGong.m](#), please replace

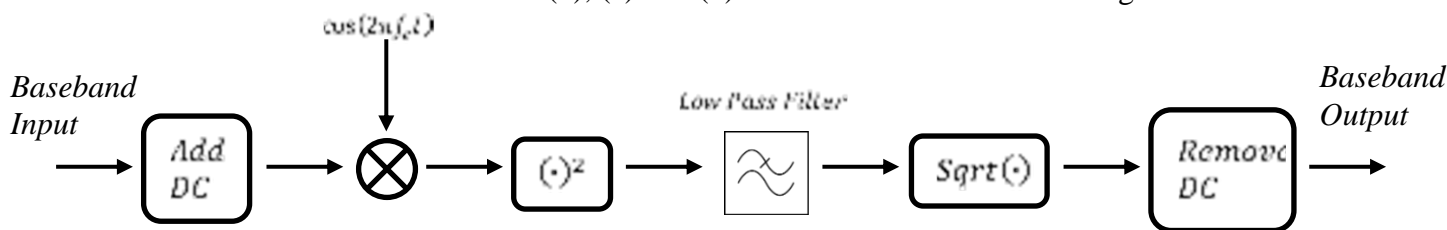
```
modulated = basebandInput .* carrier;
```

with

```
modulated = (1 + basebandInput) .* carrier;
```

- (a) In [upconvertGong.m](#), add code in the receiver to remove the DC offset added in the transmitter. The Matlab function `mean` will compute the DC (average) value of a vector. *8 points*
 - i. Compute the mean squared error between the original gong waveform and the downconverted gong waveform with the DC offset removed in [upconvertGong.m](#).
 - ii. Describe any differences that you hear between the two versions of the gong waveform.
- (b) In the [upconvertGong.m](#), replace the downconversion code with a squaring device followed by a lowpass filter. You will need to change the design of the lowpass filter. *12 points*.
 - i. Compute the mean squared error between the original gong waveform and the downconverted gong waveform for [upconvertGong.m](#) that you had modified.
 - ii. Describe any differences that you hear between the two versions of the gong waveform.
 - iii. What happened to the principal frequencies in the gong signal after squaring the upconverted signal? You might try using the `plotspec` command from the JSK book.
- (c) The squaring block distorts the amplitude of the input signal. To your solution in part (b), add a square root operation after the lowpass filter. *8 points*.
 - i. Compute the mean squared error between the original gong waveform and the downconverted gong waveform for [upconvertGong.m](#) that you had modified.
 - ii. Describe any differences that you hear between the two versions of the gong waveform.
- (d) After the square root block, remove the DC (average) value. *3 points*.
 - i. Compute the mean squared error between the original gong waveform and the downconverted gong waveform for [upconvertGong.m](#) that you had modified.

The transmitter and alternate receiver for (b), (c) and (d) are described in the block diagram below:



Bandwidth B

$$2B < f_{cutoff} < 2f_c - 2B$$

This problem is inspired by JSK, Exercise 3.19 on page 52.