Assignments

(Version: 2025-05-19)

1 General Information

With the possible exception of Assignment 1 (which is mostly review material), each of the assignments in the course will take a **significant amount of time to complete**. Therefore, students are **very strongly advised** to start working on each assignment **well in advance** of the submission deadline. To assist students in being able to start working on each problem as soon as possible, the general topic covered by each problem is indicated in square brackets after the problem number. In this way, a student can more easily determine whether they should be able to solve a problem based on the material covered in the lectures so far. Some assignments are partitioned into multiple parts (e.g., parts A and B). For such assignments, the submission deadlines for different parts may be different.

Submission Deadlines

The submission deadlines for assignments will be posted on the course website and/or the Brightspace site.

Policy on Late/Incomplete Assignments

Late assignments will not be accepted and will receive a mark of zero. Incomplete assignments will be accepted, however. So, it is much better to submit an incomplete assignment on time than a complete assignment that is late.

Assignment Submissions

The following information **must be included on the first page** of each assignment submission:

1. the student's full name (with the family name written either **last or in all-capital-letters** in order to distinguish it from the given names);
2. the student ID; and
3. the student's **lecture section** (e.g., A01, A02, etc.).

Unless explicitly indicated otherwise, for each of the MATLAB problems in the assignments, the student must include the following in their assignment submission:

1. a source listing of any code written; and
2. a copy of any output/results produced by MATLAB, such as graphs or numerical results.

Assignment Problems

For the most part, the problems that constitute each assignment are simply identified by number (e.g., A.1, 2.1, etc.). Problems identified only by number can be found in the textbook. More specifically, Problem *x.y* can be found in the textbook at the end of chapter/appendix *x*.

Additional Remarks Concerning MATLAB

Students who are less familiar with MATLAB are **strongly encouraged** to read the MATLAB appendix (i.e., Appendix D) of the textbook before attempting any of the MATLAB problems in this course, as this will very likely save such students a considerable amount of time in the long run.

2 Assignment 1 — Complex Analysis (Mostly Review)

Before starting work on this assignment, it is **critically important** that the student carefully read Section 1 (titled "General Information"), which starts on page 1 of this document.

Regular Problems

- ◇ A.1 c **`[convert to Cartesian form]`**
- ◇ A.2 b d **`[convert to polar form, principal argument]`**
- ◇ A.3 a b f g **`[complex arithmetic]`**
- ◇ A.4 b e **`[properties of complex numbers]`**
- ◇ A.5 c f **`[magnitude/argument]`**
- ◇ A.6 b **`[Euler's relation]`**
- ◇ A.11 c d **`[continuity, differentiability, analyticity]`**
- ◇ A.13 b c **`[poles/zeros]`**

MATLAB Problems

This assignment has no MATLAB problems.

3 Assignment 2 — Preliminaries and Continuous-Time Signals and Systems

Before starting work on this assignment, it is **critically important** that the student carefully read Section 1 (titled "General Information"), which starts on page 1 of this document.

3.1 Part A

Regular Problems

- ◇ 2.1 a b c d **`[notation]`**
- ◇ 2.2 a b c d e f g h i **`[notation]`**
- ◇ 3.1 f **`[time/amplitude transformations]`**
- ◇ 3.2 a **`[time tranformations]`**
- ◇ 3.4 a b c d **`[time/amplitude transformations]`**
- ◇ 3.6 e f g **`[periodicity]`**
- ◇ 3.9 c d **`[even/odd symmetry]`**
- ◇ 3.10 b **`[symmetry and sums/products]`**
- ◇ 3.17 b **`[even/odd, causal, signal transformations]`**
- ◇ 3.17 c **`[causal, even/odd decomposition, signal transformations, unit-step function]`**
- ◇ 3.20 a b c f **`[properties of delta function]`**

MATLAB Problems

- ◇ D.1 a b c d e **`[MATLAB identifiers]`**
- ◇ D.2 a b c d **`[MATLAB expressions]`**

3.2 Part B

Regular Problems

- ◇ 3.22 c **`[representations using unit-step function]`**
- ◇ 3.24 d g **`[memoryless]`**
- ◇ 3.25 b f **`[causal]`**
- ◇ 3.26 b e **`[invertible]`**
- ◇ 3.27 d e **`[BIBO stable]`**
- ◇ 3.28 b d **`[time invariant]`**
- ◇ 3.29 b e **`[linear]`**
- ◇ 3.33 b **`[eigenfunctions]`**

MATLAB Problems

- ◇ D.3 **`[temperature conversion, looping]`**
- ◇ D.4 a b c **`[write unit-step function]`**
- ◇ 3.201 a f **`[element-wise operations, case collapsing]`**

4 Assignment 3 — Continuous-Time LTI Systems

Before starting work on this assignment, it is **critically important** that the student carefully read Section 1 (titled "General Information"), which starts on page 1 of this document.

4.1 Part A

Regular Problems

◇ 4.1 e f `[compute convolution]`
◇ 4.3 b g `[compute convolution]`
◇ 4.5 `[manipulation of expressions involving convolution]`
◇ 4.6 a `[convolution property proof]`
◇ 4.9 `[meaning of LTI]`

MATLAB Problems

◇ D.5 `[plot, abs, angle, complex numbers]`

4.2 Part B

Regular Problems

◇ 4.11 a b c `[find impulse response]`
◇ 4.12 a b `[impulse response and series/parallel interconnection]`
◇ 4.13 b c `[convolution, impulse response, system interconnection]`
◇ 4.14 a f g `[causality, memory]`
◇ 4.15 a b `[BIBO stability]`
◇ 4.16 `[inverse system]`
◇ 4.17 a `[system function, eigenfunction]`

MATLAB Problems

◇ D.8 a b `[graphic patterns]`

5 Assignment 4 — Continuous-Time Fourier Series

Before starting work on this assignment, it is **critically important** that the student carefully read Section 1 (titled "General Information"), which starts on page 1 of this document.

Regular Problems

- ◇ 5.1 a c `[find Fourier series]`
- ◇ 5.2 c `[find Fourier series]`
- ◇ 5.3 a `[find Fourier series]`
- ◇ 5.7 b `[odd harmonic proof]`
- ◇ 5.9 `[find/plot frequency spectrum]`
- ◇ 5.10 `[filtering]`

MATLAB Problems

- ◇ 5.201 a b c `[Fourier series convergence]` [**Note:** It is not a requirement that you use the Symbolic Math Toolbox for this problem. If, however, you do use this approach, Appendix D of the textbook has some helpful material on the MATLAB Symbolic Math Toolbox (e.g., functions such as `symsum`, `sym`, `subs`, etc.). Refer to the section titled "Symbolic Math Toolbox".]

6 Assignment 5 — Continuous-Time Fourier Transform

Before starting work on this assignment, it is **critically important** that the student carefully read Section 1 (titled "General Information"), which starts on page 1 of this document.

6.1 Part A

Regular Problems

- ◇ 6.1 c d **`[find Fourier transform by first principles]`**
- ◇ 6.3 c d e f g **`[find Fourier transform]`**
- ◇ 6.4 a b c d e f **`[find Fourier transform]`**
- ◇ 6.5 a **`[find Fourier transform of periodic signal]`**
- ◇ 6.10 a **`[find frequency/magnitude/phase spectrum]`**

MATLAB Problems

This part of the assignment has no MATLAB problems.

6.2 Part B

Regular Problems

- ◇ 6.14 b **`[differential equation to frequency response]`**
- ◇ 6.15 b **`[frequency response to differential equation]`**
- ◇ 6.16 a **`[filtering]`**
- ◇ 6.17 a b c d **`[circuit analysis, frequency response, impulse response]`**
- ◇ 6.24 a b **`[amplitude modulation]`**
- ◇ 6.26 a b c **`[sampling]`**
- ◇ 6.27 a b **`[sampling]`**

MATLAB Problems

- ◇ 6.201 a b c **`[calculate frequency response]`**
- ◇ 6.203 a b c d **`[filters]`** [**Hint:** The MATLAB appendix (i.e., Appendix D) in the textbook has some examples of how to use the `butter` and `besself` functions. Refer to the section titled "Signal Processing" and its associated subsections for more information. In particular, the specific pages of relevance can be found by looking up the terms "Butterworth filter" and "Bessel filter" in the textbook index. To compute the frequency response from the coefficient vectors obtained from the `butter` and `besself` functions, you can use the `freqw` function developed in Problem 6.101. Alternatively, the `freqs` function can be used to calculate the frequency responses of the filters from the coefficient vectors returned by the `butter` and `besself` functions.]
- ◇ Problem M.1:
  Background: The sampling theorem states that a (bandlimited) continuous-time signal can be uniquely/unambiguously represented by its samples. Therefore, all of the operations that we can apply to a continuous-time signal can be converted into equivalent operations on their samples. When processing signals inside of a computer, this is always how things are done. That is, we operate on the samples of a continuous-time signal instead of the original continuous-time signal directly. In this problem, you will experiment with some code that processes continuous-time signals by performing equivalent operations on their samples.
  Comment on Negative Frequencies: In this problem (and the associated MATLAB code), when dealing with frequency spectra, we only concern ourselves with nonnegative frequencies since real-valued signals always have even/odd symmetry in their magnitude/phase spectra, making the half of the spectra for negative frequencies redundant.
  Problem: Download the `audioDemo.zip` Zip archive from the "Assignments" section of the course web-site home page. This archive contains several MATLAB source files. Extract the contents of the Zip file using the unzip

command (i.e., “unzip audioDemo.zip”) and place the extracted files in a directory in which MATLAB searches for M-files. The main program file is called `audioDemo.m`. Examine this file in some detail as it provides a basic template for doing this problem. That is, to do this problem, you will only need to comment/uncomment or make very trivial changes to various lines in this file. You should not need to change any of the code except the code in `audioDemo.m`.

(a) For the `train` audio signal, use the template program provided (in `audioDemo.m`) to plot the signal and its frequency spectrum as well as to play the signal on the audio device (i.e., speaker). Make a hardcopy of the plot of the signal and its frequency spectrum. By examining the frequency spectrum, identify at which three (nonnegative) frequencies the train whistle has the most information/energy.

(b) For the `handel` audio signal, use the template program provided (in `audioDemo.m`) to plot the signal and its frequency spectrum as well as to play the signal on the audio device. Make a hardcopy of the plot of the signal and its frequency spectrum. Then, do the same thing for the `noisyHandel` audio signal, which is essentially the `handel` signal with a significant amount of noise added for (nonnegative) frequencies in the range $[3000, 3500]$ Hz. Identify the noise on the plot of the frequency spectrum. Apply a bandstop filter with a stopband corresponding to (nonnegative) frequencies in the range $[2950, 3550]$ Hz to the noisy signal. (Note that a bandstop filter is like a bandpass filter, except that instead of passing frequencies in a certain range, frequencies in a certain range are eliminated.) Again, plot the signal spectrum and play the signal on the audio device. Describe what effect the filter had on the signal being processed. **`[filtering]`**

Note: Since the MATLAB source code is provided for this problem, it is not necessary to include a copy of this source code in your assignment submission.

7 Assignment 6 — Laplace Transform

Before starting work on this assignment, it is **critically important** that the student carefully read Section 1 (titled "General Information"), which starts on page 1 of this document.

7.1 Part A

Regular Problems

- ◇ 7.1 c **`[find Laplace transform by first principles]`**
- ◇ 7.2 b c d e **`[find Laplace transform]`**
- ◇ 7.4 a **`[find Laplace transform (from graph)]`**
- ◇ 7.5 e **`[find Laplace transform]`**
- ◇ 7.6 a b **`[initial/final value theorem]`**
- ◇ 7.10 d **`[find inverse Laplace transform]`**
- ◇ 7.12 **`[find inverse Laplace transform]`**

MATLAB Problems

This part of the assignment has no MATLAB problems.

7.2 Part B

Regular Problems

- ◇ 7.13 a **`[system function to differential equation]`**
- ◇ 7.14 a **`[differential equation to system function]`**
- ◇ 7.16 a b **`[stability analysis]`**
- ◇ 7.17 a b c d **`[circuit analysis, stability analysis, step response]`**
- ◇ 7.18 **`[inverse systems and system function]`**
- ◇ 7.20 **`[communication systems, equalization]`**
- ◇ 7.21 a **`[solve differential equation]`**
- ◇ 7.22 a b **`[solve differential equation for circuit]`**

MATLAB Problems

- ◇ 7.201 a b **`[stability analysis]`** [**Hint:** The `roots` function might be helpful.]
- ◇ 7.202 a b **`[impulse/step response]`** [**Note:** Appendix D of the textbook has some information on the MATLAB Signal Processing Toolbox (e.g., functions such as `tf`, `impulse`, `step`, etc.). Refer to the section titled "Signal Processing" and its associated subsections.]

