

**Matlab Project II**

Fall 2004

**Issued:** Tuesday, October 19, 2004**Individual part due:** Tuesday, November 2, 2004**Team part due:** Thursday, November 11, 2004

The purpose of this project is to give you some practice with the discrete time Fourier transform and to provide a brief introduction to the `fft` command. We will study how the `fft` works during the lectures on 10/26 and 10/28. You do not need to know the details of the inner-workings of the `fft` command to do this project, however. This project will also give you additional practice in working in teams. You will work with your fellow in-class group members to complete the second part of this project.

**Part 1: Individual work** (worth 50% of grade)

Each student should do the Basic and Intermediate problems in Section 5.2 of *Computer Explorations in Signals and Systems* by Buck, Daniel, & Singer. The writeup for this part should include answers to all of the questions in the book and any other observations you make as you complete the exercises. Each student will turn in their own writeup for this part by November 2, 2004.

Each student must do his or her own work on this part of the project. Copied plots or code are not allowed. You may ask other students for advice. As stated in the guidelines given in the ECE 410 course information packet, you should identify any students you collaborate with.

**Part 2: Group work** (worth 50% of grade)

Each group should do the Advanced problems in Section 5.2 of *Computer Explorations in Signals and Systems* by Buck, Daniel, & Singer. In addition to these exercises, each group will develop a plan for testing the `ttdecode` function to ensure that the function will work on other signals besides the ones provided for the project. This test plan should be implemented and the results documented. Each group will turn in one writeup for this part by November 11, 2004. The writeup for this part should include the following:

- A typed memo containing:
  - answers to all of the questions in the book.
  - a description of how the `ttdecode` function works
  - a discussion of the testing plan and the results
- An electronic copy of the function `ttdecode`. Bring this to class on a floppy disk or a USB flash drive or email to `kwage@gmu.edu` before class time.
- 2 overhead slides
  - Slide 1 describes the testing plan and the results
  - Slide 2 summarizes what was learned during this project and discusses what the group would do differently if they had to do this again.

During class on November 11, 2004 each group will be given 5 minutes to present their results. One person from each group will be randomly selected (by a roll of dice) to present their group's 2 slides. The 5-minute time limit will be strictly enforced. After all groups have presented their slides, the performance of each group's function will be benchmarked using a set of signals designed by the teaching staff.

Report guidelines:

Writeups must include all of the analytical (*i.e.*, pencil/paper) work, Matlab plots and code, and relevant explanations. A list of guidelines for preparing the writeup of this project are given below.

- The report for part 1, may be handwritten (neatly) or typed. The memo/report for part 2 must be typed.
- All pages of reports and memos must be numbered.
- All plots must be neatly annotated with x-axis and y-axis labels and a title. Any graph not labeled will be considered not handed in.
- I will not spend time trying to figure out which graphs are for which problems. When referring to plots in the text, I recommend doing at least one of the following:
  - use figure numbers, e.g., “Figure 1 is a plot of the signal  $x[n]$ .”
  - cite the page number they are on, e.g., “The figure at the top of page 4 is a plot of  $x[n]$ .”
- All Matlab code must be well-documented and should be included in an appendix at the end of the report.

Additional instructions:

- For better sound, you may want to use the `soundsc` command, which automatically scales the sound vector before playing it. This will play the sound as loud as possible without clipping.
- The command `fft(x, N)` produces  $N$  samples of the DT Fourier Transform, evenly spaced between 0 and  $2\pi$ . In other words the spacing between samples is  $\frac{2\pi}{N}$ , and the first sample is at 0. If you use the `fftshift` command, as suggested in part e, the samples will be reordered to start at  $\omega = -\pi$ . The spacing between samples is the same, *i.e.*,  $\frac{2\pi}{N}$ . If you want to plot the output of `fftshift`, you will need to define another `omega` vector, one that starts at  $-\pi$  and ends at  $\pi - \frac{2\pi}{N}$ .