Digital Signal Processing (18-491/18-691)
Spring Semester, 2025

Problem Set 9

Issued: 3/27/25

Due: 4/4/25 0100 via Gradescope

Note: Although the nominal due date is April 4 for this problem set, students may turn it in until April 7 at 0100 without penalty because of the CMU Carnival. Note that the second exam will take place two days later, though.

Reminder: Quiz 2 will take place on April 9, in class. The exam coverage will be through this problem set, with questions focussing on material since the first quiz, including Fourier representation of periodic functions, the DFT, circular and linear convolution, OLA and OLS, basic FFT structures, digital filter implementation, and IIR digital filter design.

Reading: Most of the problems on this problem set concern the design of IIR filters, reviewing the impulse invariance and especially the bilinear transformation approaches. This material follows the material in OSYP Secs. 7.0 through 7.4 as well as the supplementary class handout on IIR filter design that is also available on the Web. This problem set reviews these design procedures using both manual design procedures and using computer-aided designs with MATLAB . In working the problems on this set, please be sure to review the handout on IIR filter design using Butterworth prototypes and the bilinear transformation method. As you know, we are continuing with FIR filter design, following the material in OSYP Secs. 7.5 through 7.9.

Problem 9.1: Problem 7.26 in OSYP.

Problem 9.2: In this problem we will implement a discrete-time lowpass filter using the bilinear transformation method, and using a *Butterworth* prototype filter.

Here are the specifications for the filter to be designed:

- Passband cutoff frequency: $\omega_p = 0.5\pi$
- Stopband cutoff frequency: $\omega_s = 0.75\pi$
- Passband ripple[1]: $-2 \text{ dB} \le |H(e^{j\omega})| \le 0 \text{ dB},\ 0 \le |\omega| \le \omega_p$

[1] Note that the values of passband and stopband attenuation are given in dB. If x has the dimensions of amplitude or magnitude (as opposed to energy or power), the value of x expressed in dB would be $20 \log_{10}(x)$.

- Stopband attenuation: $|H(e^{j\omega})| \leq -20$ dB, $\omega_s \leq |\omega| \leq \pi$

Using the bilinear transformation method and Butterworth continuous-time filter prototypes, determine the following:

(a) What are the passband and stopband critical frequencies (Ω_p and Ω_s) in the continuous-time domain?

(b) What is the order N of the filter?

(c) What is the continuous-time form of the filter's transfer function, $H_c(s)$?

(d) What is the discrete-time form of the filter's transfer function, $H(z)$?

(e) Draw a cascade-form or parallel-form implementation of this filter (using first and second-order Direct Form II filter structures). You may use MATLAB to help with the calculation in this part of the problem.

(f) Draw a cascade-form or parallel-form implementation of this filter (using first and second-order DirectForm II filter structures). You may use MATLAB to help with the calculations in this part of the problem.

Problem 9.3: It is claimed that a discrete-time highpass filter can be obtained from a continuous-time lowpass filter by the following transformation, which is similar to (but not quite identical to) the conventional bilinear transformation:

$$s = \frac{2}{T}\frac{1+z^{-1}}{1-z^{-1}}$$

(a) Can this really be true? Show that the transformation above maps the $j\Omega$-axis of the s-plane onto the unit circle of the z-plane.

(b) Show that if $H_c(s)$ is a rational function with all its poles inside the left-half s-plane, then $H(z)$ will be a rational function with all its poles inside the unit circle of the z-plane.

(c) If the transformation above is used to map a prototype continuous-time filter into a discrete-time filter, what is the frequency relationship obtained between ω, the discrete-time frequency variable, and Ω, the continuous-time frequency variable? Sketch the function relating Ω to ω.

(d) Now we would like to apply the relation above to the design of a discrete-time high-pass filter from a continuous-time lowpass filter. Assume that the discrete-time highpass filter that you are designing is intended to be a part of a system that samples continuous-time signals at a rate of 16 kHz, converts them to discrete time using an ideal D/C converter (with an appropriate anti-aliasing lowpass filter as in OSYP Fig. 4.14), passes the discrete-time signals through the discrete-time highpass filter, and then converts the discrete-time signal back to continuous time, again at a rate of 16 kHz.

Assume that the overall system is intended to have the following specifications (expressed in *continuous time*):

- Passband frequencies: 6 to 8 kHz
- Stopband frequencies: 0 to 5.5 kHz
- Passband ripple: -1 dB $\leq |H(j\Omega)| \leq 0$ dB, $\Omega_p \leq |\Omega| \leq 8$ kHz
- Stopband ripple: $|H(j\Omega)| \leq -60$ dB, $|\Omega| \leq \Omega_s$

1. What are the corresponding specifications of the highpass filter in *discrete time*?

2. Assume that the filter would be implemented using the bilinear transform and a continuous-time lowpass Butterworth prototype filter. Using the parameter value $2/T = 1$, what are the **order and cutoff frequency** of the filter that you would need to meet the specifications? **Note:** You do not need to obtain the complete z-transform of the continuous-time prototype IIR filter; only the order and cutoff frequency of the filter is requested.

Problem 9.4: Problem 7.48 in OSYP, parts (a) through (e)

MATLAB Problems

Note: The signal processing package in MATLAB itself contains a number of functions that facilitate the design of IIR digital filters including the following:

- Functions that perform complete filter design from discrete-time specifications: `butter, cheby1, cheby2, ellip`
- Functions that estimate the order of the prototype continuous-time filter that will be needed or realize a particular set of specifications: `buttord, cheb1ord, cheb2ord, ellipord`
- Functions that enable you to design "by hand" the continuous-time prototype filters: `buttap, cheb1ap, cheb2ap, ellipap`
- Functions that convert continuous-time filters into discrete-time filters: `bilinear, impinvar`
- Functions that perform frequency transformations: `lp2lp, lp2hp, lp2bp, lp2bs`

- The design aid `filterDesigner` that displays responses and parameters for IIR and FIR filters.

Keep in mind that the MATLAB routines `poly` and `roots` can be extremely helpful in multiplying and factoring polynomials. You are strongly encouraged to look over all of the help files for these functions to see how they work. In working the MATLAB problems, turn in a printout of your results, a copy of the MATLAB space code you developed to work the problem, as well as any additional comments you'd like to add.

Problem C9.1: In this problem we will use various MATLAB routines to check your work in designing the Butterworth filter by hand in Problems 9.2.

(a) Use the MATLAB routine `buttord` to confirm the order and critical frequency of the prototype filter that you developed in part (a) of Problem 9.2. The order produced by `buttord` should be the same as you have obtained, but the critical frequency may be different. If so, explain whether you consider the critical frequency produced by `buttord` to be valid in terms of your calculations.

(b) Use the MATLAB routine `butter` with the `‘s’` option to confirm the continuous-time prototype transfer function you developed in part (c) of Problem 9.2.

(c) Use the MATLAB routine `butter` with the `‘s’` option combined with the routine `bilinear`, or use the MATLAB routine butter directly to verify the discrete-time transfer function that you developed in part (d) of Problem 9.2.

Problem C9.2:

In this problem we will do a bit of introductory exploration of the MATLAB filter design tool `filterDesigner`. This is probably the most useful general utility for solving real filter problems with MATLAB .

The opening screen (after the Tip of the Day) has a GUI with a lot of filter design parameters. Most of them should be fairly obvious. A few comments:

- For designing filters using OYSP notation, use normalized frequency. (Under `Units` for `Frequency Specifications`, use `Normalized (0 to 1)`). Under this system, discrete-time frequency $\omega = 0.45\pi$ would be entered simply as 0.45.
- Similarly, under `Magnitude Specifications` use `dB` for `Units`. Attenuation is entered as a positive number. For example, the specs for the filter in Problem 9.2 would be entered as `.5, .75, 2`, and `20` for `wpass, wstop, Apass`, and `Astop`, respectively.
- Explore the displays that are provided through the icons on the top row of the panel which include magnitude, phase, group delay, phase delay, pole-zero locations, impulse response, and step response, among other things.
- Also note that you can write filter coefficients to a file in various formats including MATLAB code through the `Export` command under the `File` menu. Also note that there are various GUIs for implementing the filters along the lower left border of the panel.

Consider a lowpass filter with the specifications

- Passband cutoff frequency: $\omega_p = 0.65\pi$
- Stopband cutoff frequency: $\omega_s = 0.75\pi$
- Passband ripple: $-1 \text{ dB } \leq |H(e^{j\omega})| \leq 1 \text{ dB}, \; 0 \leq |\omega| \leq \omega_p$
- Stopband attenuation: $|H(e^{j\omega})| \leq -70 \text{ dB}, \; \omega_s \leq |\omega| \leq \pi$

What is the order of the discrete-time filter that is needed to realize these specs using the following types of IIR filters?

(a) Butterworth

(b) Chebyshev Type I

(c) Chebyshev Type II

(d) Elliptical