SPR401T GROUP PROJECT

DUE DATE: WEDNESDAY 31st OCTOBER 2012 AT 12:00 PM

1. ORGANISATION OF THE PROJECT

You must work in groups of 4 people. You can organise yourselves in groups as you please.

A Project report of 15 pages maximum (Project A and Project B included) with attached MATLAB .m files must be submitted by email before 12:00 p.m. on the 31st October 2012. **N.B: No Late Submissions will be tolerated!**

Your work must be original. Any evidence of copying between two groups will result in a zero mark for both.

Your work must be properly referenced (IEEE referencing style). It should not be a "copy and paste" job. Any evidence of plagiarism will result in a zero mark.

No formal solutions to the project will be handed out to you.

2. DESCRIPTION OF THE PROJECT

PROJECT A: THEORY

Among the most popular analogue filter types, we distinguish **Butterworth filters**, **Chebychev (Type I and Type II) filters**, and **Elliptic filters**.

A.1. Write a report (2 pages maximum) that describes the characteristics of each of the low-pass versions of these filters in terms of *pass-band*, *stop-band*, and *frequency spectrum.*

A.2. These analogue filters are usually implemented digitally. There are two methods to convert the analogue designs into their digital (discrete) equivalents, namely the *bilinear transform method* and the *impulse invariant method*. Write a report (1-page maximum) that describes the two methods.

PROJECT B: PRACTICAL

Project Background [1]

The aim of this project is to introduce you to discrete-time signal processing and help you understand that it is tightly linked with analogue (continuous-time) signal processing. Discrete-time signal processing has been widely applied in the field of speech processing especially noise suppression. This project explores the application of digital filtering for noise suppression in speech. You will be required to design 4 types of low-pass Infinite Impulse Response (IIR) filters digitally, namely **a Butterworth filter**, **a Chebychev Type I filter, a Chebychev Type II filter**, and an **Elliptic filter,** using MATLAB.

In the past, Signal Processing courses were putting much focus on design techniques for both Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filters. Now that a lot of filter design algorithms are implemented in common software packages such as MATLAB, it has become unnecessary for most practitioners to learn the details of the design algorithms. However, abandoning the design task to a software package makes it very important to understand the properties of different types of optimal filters, the meanings of the design parameters, and some of the trade-offs between filter types.

In the design of IIR filters, MATLAB takes specifications for ripple magnitude and stopband attenuation as input parameters. A specification of ripple magnitude δ_{IIR} yields an IIR filter design with a maximum passband gain of unity (or 0 dB) and a minimum passband gain of $10^{-\delta_{IIR}/20}$ (or $-\delta_{IIR}$ dB). A specification of stopband attenuation ξ_{IIR} yields an IIR filter design with a maximum stopband gain of $10^{-\xi_{IIR}/20}$ (or $-\xi_{IIR}$ dB).

To understand IIR filters designed with MATLAB, one must find a correspondence between the desired specifications and the input parameters to the filter design functions. Let Gpb_{max} (dB), Gpb_{min} (dB), and Gsb_{max} (dB) refer to the desired maximum passband gain, minimum passband gain, and maximum stopband gain, respectively. In order to have free control over each of these parameters, let k_{IIR} be the linear scaling factor which is used to normalize the IIR designs.

Project Specifications [1]

Before starting your design, the file projIB.mat will need to be first loaded into MATLAB. After downloading this file from myTUTor, copy it into your Matlab working directory and type load projIB.

This project is concerned with designing low-pass IIR filters for the removal of high-pass noise from a speech signal. The noisy signal is stored in the variable "noisy" and was sampled at 44100 Hz (Sampling frequency f_S = 44100 Hz). It consists of a summation of a speech signal which was band-limited at 4 kHz using a low-pass filter with a very narrow transition band, and a noise signal which was filtered at 4 kHz using a high-pass filter with a very narrow transition band. To filter the noise you must design low-pass filters filter with the following parameters:

- Passband edge: 2500 Hz.
- Stopband edge: 4000 Hz.
- Maximum gain in the passband Gpb_{max}: 40 dB.
- Minimum gain in the passband Gpb_{min}: 37 dB.
- Maximum gain in the stopband Gsb_{max}: −55 dB.

Project Questions [1]

B.1. Find δ_{IIR}, ξ_{IIR}, and k_{IIR} in terms of Gpb_{max}, Gpb_{min}, and Gsb_{max}. These parameters will be used for the filter designs.

B.2. Design a discrete-time low-pass filter of each of the following types based on the specifications given above: Butterworth, Chebyshev Type I, Chebyshev Type II, and Elliptic.

Write a MATLAB .m script for each filter. Implement the filters according to the specs using MATLAB's built in tools (such as the command line functions *butter, cheby1, cheby2,* and *ellip*). Type *help* followed by the function (e.g.: "*help butter*") for more information.

For each of the designs, provide the following in your project B report:

a) The order of the filter. Use MATLAB's order estimation functions (e.g. *buttord, cheb1ord*…).

b) Determine the filter coefficients (Use *butter, cheby1, cheby2,* and *ellip*).

c) Plot the magnitude response (in dB) from $\omega = 0$ to $\omega = \pi$ using *freqz*. Plot a detail of the magnitude response, focusing on the passband ripple (linear scale). Plot the group delay (in samples) using *grpdelay*. (Use *subplot* to fit the three plots on the same page for each filter).

d) Plot the pole-zero diagram.

e) Plot the impulse response using *filter* and *stem* for 100 samples. (Use *subplot* to fit the pole-zero diagram and the impulse response on the same page).

f) Filter noisy using your de-noising filter (Use *filter* with the coefficients provided by your *butter*, *cheby1*, *cheby2*, and *ellip* functions. You might need to amplify your signal by scaling one of your coefficients). Plot the noisy signal before filtering and the de-noised signal after filtering.

g) Listen to the filtered and original files. How do they compare? (Use *audioplayer* and *play*).

REFERENCES

[1] Massachusetts Institute of Technology (MIT), "6.341: Discrete-Time Signal Processing - Project Assignment," Fall 2005. [Online].