EEE321 Lab Work 6

(Clearly justify all answers.)

(Due 25 December 2018 for Sec. 3, 26 December 2018 for Section 1, and 28 December 2018 for Secs. 2 and 4.)

Work on these questions as a homework first: answer the analytical parts of the questions and write the answers on a paper, write a MATLAB program (or many such programs) to perform the tasks that need computation, print your MATLAB code(s), print your computer outputs (numerical and graphic) whenever needed; the collection of all those will be your lab report. Bring your code (in a computer readable form) to the lab; transfer your code to one of the lab computers; run and show your TA the results. Answer all the questions your TAs may ask. Modify your lab report, including any modifications needed in your MATLAB codes, during the lab hours in the lab. Finally, hand your TAs the lab report prepared as described above. Your report will get a grade based on your preparedness when you come to the lab, performance of your codes in the lab (any modifications needed and conducted during the lab hours included), your answers to the oral questions during your demo(s) in the lab, and the entire content of the submitted report at the end of the lab.

****

In this work, you will design a practical IIR filter, test its frequency response, and perform some filtering operations. The IIR filter that you will design will be in the form of a difference equation; therefore, the "design" means to determine the coefficients of the difference equation.

Before you start, read the third and fourth least significant digits of your student number. If these two numbers are the same, read the third and fifth least significant digits; and continue with third and sixth, third and seventh, etc., until the two numbers are not the same. Call these two distinct number as N_1 and N_2, respectively. Find $M_1 = (N_1 + 2)$, and $M_2 = (N_2 + 2)$. Write your N_1, N_2, M_1, and M_2.

The filter specifications are as follows:

* Stable (Naturally, this is needed for all practical filters.)

* Bandpass

* Real valued $h[n]$.

* Order of the filter is $3 + N_1$.

* Cutoff frequencies are $\min\{\frac{\pi}{M_1}, \frac{\pi}{M_2}\}$ and $\max\{\frac{\pi}{M_1}, \frac{\pi}{M_2}\}$

* A stopband and a passband, which are as flat as possible, are desirable.

* Causal; $h[n]$.

Clearly write the analytical details of your design procedure. Avoid computer aided design as much as possible.

Your typical design procedure could be: i) Determine the number of poles from your filter order, ii) If you wish add some zeros iii) Noting the restrictions on the locations of poles and zeros, as a consequence of your specifications, place the poles and zeros on the z-plane iv) Write $H(z) = B(z)/A(z)$, where $B(z)$ and $A(z)$ are obtained from your poles and zeros, first in product from of first order factors, and then as a polynomial of z, then convert them to polynomials of z^{-1}. v) Write your $H(z)$, and the corresponding difference equation using your polynomial coefficients.

1- Print $h[n]$ as an array, and plot $h[n]$, for a reasonable length. Make a pole-zero plot of $H(z)$. Also plot, $H(e^{j\omega})$, for some discrete values of ω in the range $[-\pi, \pi)$, using MATLAB.

2- As you did in the previous lab work, generate the two discrete chirp signals, $x_1[n]$ and $x_2[n]$, by sampling the analog signal $x_a(t)$ as $x_i[n] = x_a(nT_i)$; $x_a(t) = \cos(\alpha t^2)$; for $x[n_1]$ the sampling rate is $T_1 = \sqrt{\frac{\pi}{\alpha 512}}$, and for $x_2[n]$ the sampling rate is $T_2 = \sqrt{\frac{\pi}{\alpha 8192}}$; note that these discrete signals are periodic; choose a finite segment of those signals in the range $n \in [0, 1023]$ and $n \in [0, 8191]$, for $x_1[n]$ and $x_1[n]$, respectively.

3- Pass your two test signals through your IIR filter, by writing a recursion to implement your difference equation, using MATLAB. Store your output arrays, $y_1[n]$ and $y_2[n]$ for a suitable duration. Discuss what the output represents. Discuss whether this procedure (passing a chirp through an LTI system) can be used to test/understand the frequency response of the system. Make sure to comment on the limitations, and precautions to relieve those limitations.

4- As you did in your previous lab work, write a separate MATLAB program to forward the signal $x_2[n]$, at a rate of one sample per T_s (T_s is as in (2)), via the system D/A converter as an analog waveform to the system loudspeaker, starting from $n = 0$, and then continuing indefinitely until the program is stopped manually. (Hint: A natural way to do this is to compute the elements of $x_2[n]$ in real time, and to pass the computed elements, one by one as they are computed, at instants separated by precise T_s seconds apart, to the loudspeaker. If this is difficult to achieve in your programming environment, an alternative is (note that $x_2[n]$ is periodic) to compute and store one period of the signal to a file (first find the period), off line, and then to retrieve this array from the file many (indefinite) times, one after another, and to forward these arrays seamlessly to the laudspeaker, by instructing the system about the sampling rate.) Thus, you are listening to $x_r(t)$; Is it equal to $x_a(t)$ in (3)? (Beware: $x_a(t)$ is not periodic; try to imagine the sound represented by $x_a(t)$.)

5- Using the same MATLAB program above, listen to $y_r(t)$, which is the analog version of $y_2[n]$ (i.e., $y_r(t)$ is the interpolated version of $y_2[n]$ by the system D/A converter using the same sampling rate as in (2) during the D/A conversion; both $y_2[n]$ and $y_r(t)$ are finite length signals). Comment on what you hear.

6- Write the cut-off frequencies of the equivalent analog system whose input is $x_a(t)$ and the output is $y_r(t)$. Plot $|H_{eq}(j\omega)|$ using MATLAB.

7- Find out some music stored digitally; find out the sampling rate used during recording. Store a finite portion of the music as an array, and then, pass it through your filter; store the output, as well. Listen

to the input, and then the output, at the same rate. Comment on the results, and discuss. (Hint: Your MATLAB may open and store an array from a music file, if the file is one of the standard audio files.)

8- Repeat (6) by digitizing your voice as you speak, using a microphone and the system A/D converter. (Hint: First store your digitized voice for a few seconds, as a standard audio file; then your MATLAB may be able to open this file and generate an array from it.)