Audio EQ / Spectral Analysis

Assignment Due Date: Matlab code on Wednesday April 22, 10 a.m. via TURN-IN and code/plots in class on the same day.

- No traditional report required. In class, submit the following hard copy stapled as one document in this order: fireq.m code, plots generated by fireq, iireq.m code, plots generated by iireq, eve.m code, optional plots or other hard copy output generated by eve. Document your code.
- TURN-IN the following files: fireq.m, iireq.m, and eve.m. (Lower-case names please.) See web page for instructions on file submission.
- Work individually.

1. 10-Band Graphic EQ system

Write FIR-filter-based and IIR-filter-based Matlab functions that simulate the POP and DANCE audio tone-shaping graphic EQ (equalizer) systems shown below. Each design will be judged by viewing a plot of dB gain vs. frequency.

FIREQ (EQ using FIR filters)

Method:
- Design the FIR EQ system for a music sampling rate of 44.1 kHz.
- The EQ processor consists of a bank of ten digital bandpass filters connected in parallel. In an actual implementation the input signal x(n) is passed through each filter; the outputs of the ten filters are then added to give the EQ’d signal y(n). The frequency response of the overall system is the sum of the ten individual responses.
- Filter 1 has a passband of 31.25-62.5Hz and a gain factor determined by the slider dB setting in the diagram. The pass band of Filter 2 is 62.5-125Hz, etc. Filter 10 is a highpass filter that passes 16-22.05kHz.
- Write your own code to define the FIR filter b-coefficients for each bandpass filter and the highpass filter. Parameters to choose are filter length and window type.
- You must test the EQ system by first computing the impulse response (use FILTER command) of the overall system, then use FFT (not FREQZ) to compute the frequency response. Useful Matlab functions: SEMILOGX, UNWRAP, FILTER, FFT, LOG10
- Remember, this EQ system will consist of ten different FIR filters (9 BP + 1 HP) connected in parallel.

Deliverables:

fireq.m - a Matlab function called fireq in an m-file of the same name. The function call will be, for the case of the pop EQ profile:

fireq('pop')

This function will result in (only) the following outputs:
- Plot the total gain response of the ‘pop’ EQ system on a dB vs. log frequency (Hz) scale.
- Plot the total phase response of the system (phase in radians and linear frequency in Hz) below the above plot.
- See the attached example for a ROCK EQ profile.

The function will accept ‘pop’ and ‘dance’ as an input argument, and function accordingly. Here is a hint on passing the profile name through to your function:

function fireq(profile)
    (code here)
    if strcmpi(profile,'pop')

1 These EQ profiles are from iTunes. Google iTunes EQ if you want background information.
2 Lesson 18 gave a preview with some examples. Lessons 35 & 36 provide more details.
else if strcmpi(profile,'dance')
  (code here)
end
end

Points awarded for:

- Shape of EQ gain and phase profile
- Graph quality (axis range, axis labels, title, grid etc.)
- Economy in FIR order
- Code documentation
- Economy in execution time

**IIREQ (EQ using IIR filters)**

Method:

- Matlab includes a variety of digital IIR designs, such as Butterworth (BUTTER), Chebyshev (CHEBY1, CHEBY2), and Elliptic (ELLIP). The Elliptic filter is the most flexible and optimized of this group. Select filter parameters experimentally. The maximum filter order should be in the range 8-16. The IIR filters are not linear phase, although they come close over a limited frequency range. (Examine this yourself.)
- To implement the bandpass filter bank, you must use a single set of bandpass filter coefficients \{b,a\} in conjunction with decimation and interpolation. See the Appendix for more instructions on implementing the IIR bandpass filter bank. The 16-22.05kHz highpass filter will be separate and have its own set of filter coefficients.
- Use built-in Matlab functions such as ELLIP to return the filter \{b,a\} coefficients. There will be two sets of \{b,a\} coefficients for the complete system: one for the BP filter bank and one for the remaining HP filter.
- You will test the EQ system by first computing the impulse response (use FILTER command) of the overall system, then use FFT (not FREQZ) to compute the frequency response.
- To summarize, ~90% of this EQ system will consist of nine parallel operations, each of which implements one BP channel using \{Decimate → BP IIR → Interpolate\} as shown in the Appendix. A tenth parallel operation implements the HP filter straightforwardly, without any resampling.

**Deliverables**

- **iireq.m** – same as above, except using IIR filters.

Points awarded for:

- Shape of EQ gain and phase profile
- Graph quality (axis range, axis labels, title, grid etc.)
- Economy in IIR order
- Code documentation
- Economy in execution time
iTunes® EQ Profiles
2. FFT-based Spectral Analysis

Special agent Eve Zadroppa used a phone wiretap to intercept the signal wiretap.wav. The touch-tone signal appears to contain a sequence of five digits (not a zip code btw), perhaps a coded message. She suspects that the decoded message will reveal the name of a famous American.

Decoding the message will be tricky: there is significant static noise in the background and those dastardly enemy agents have corrupted each digit by adding another tone, so she can’t decode the five digit sequence by simply listening to the signal. Furthermore, the interfering tone appears to be different for each digit. (Blimey!)

Naturally, Eve doesn’t panic. She is highly trained in FFT spectral analysis and Google-savvy. Eve will win the day!! (Although she’d like your help please.)

Deliverables

- **eve.m** – a Matlab function that will decode wiretap.wav into a sequence of five single-digit numbers. The last two lines of the code will print out the decoded five-digit sequence (e.g. 3 2 8 5 5) and the name of the famous American. The Matlab code is not required to figure out the name from the sequence of numbers: that’s up to you (and Eve). My call to the function will be: eve(x) where x is the intercepted signal contained in wiretap.wav. You are allowed to use the FFT and general Matlab commands but no other significant built-in DSP functions apart from FILTER. And, no WAVEREAD commands in your code please.

---

3 One of Eve’s favourite sites is [http://hyperphysics.phy-astr.gsu.edu/hbase/sound/telton.html](http://hyperphysics.phy-astr.gsu.edu/hbase/sound/telton.html)
IIR filters designed with a very small passband (in $\omega$-space or normalized-frequency-space) can be sensitive to coefficient round off. This is depicted here:

Each of the above bandpass filters spans an octave. Since the higher octave filter is easy to realize (for realistic filter orders), one solution is to move the lower octave band up to a fixed higher octave band in $\omega$-space, apply a bandpass filter with fixed bandwidth $\omega_1$ to $\omega_2$, then move the band back down to its original location. (See system diagram below.)

Principles of the decimator and interpolator are discussed in Lesson 40.\textsuperscript{4} To gain understanding of this approach, I suggest you write Matlab code to implement the system below for $L = 2$. Use the RESAMPLE command instead of DECIMATE and INTERP. Find and plot the frequency response of this system. (This system is not strictly LTI. However, with $x(n)$ set as an impulse I was able to find the overall system impulse response and hence the frequency response.)

The exercise with $L = 2$ should give you enough of a start to understand how to design and implement the IIR EQ system.

\textsuperscript{4} To be covered earlier than the scheduled date.
SAMPLE EQ PROFILES FOR ‘ROCK’ EQ SETTING

FIR

If I was to do this graph over, I would scale the y-axis of the gain plot so that it covered -4dB (or -6dB) to +6dB since the range from 6-12 dB is unused in this EQ profile.

EQ Gain & Phase Response for iTunes(R) Rock Music

IIR

Again, I should have scaled the y-axis from -6dB to +6 (or +8) dB.

Phase response not shown for IIR EQ system in this example.