Lab 10 – Digital filter design

Objectives: In this lab we will design a few FIR and IIR filters and explore the `filterDesigner` graphical tool in MATLAB.

10.1 Low-pass FIR filter design using windows

In this **script** we will design a low-pass FIR filter which has linear-phase and length N (odd) using the method of windows. The ideal low-pass filter frequency response and the LPF with linear phase version are respectively given by,

$$H_{\mathrm{LPF}}(e^{j\omega}) = \begin{cases} 1, & |\omega| \le \frac{\pi}{7} \\ 0, & \frac{\pi}{7} \le |\omega| < \pi \end{cases} \qquad H_d(e^{j\omega}) = \begin{cases} e^{-j\omega n_c}, & |\omega| \le \frac{\pi}{7} \\ 0, & \frac{\pi}{7} \le |\omega| < \pi \end{cases}$$

What are the corresponding impulse responses $h_{\mathrm{LPF}}[n]$ and $h_d[n]$?

Set $n_c = \frac{N-1}{2}$

(a) In the method of windows, to create a causal linear-phase FIR filter, we define

$$h[n] = h_d[n]\, w[n]$$

where $w[n]$ is a window of length N. Determine the coefficients of a 51-tap (i.e. N = 51) filter based on the window method using a rectangular window for $w[n]$.

(b) Compute a 1001-point DFT of $h[n]$ to approximate its Fourier transform $H(e^{j\omega})$. Plot the filter coefficients $h[n]$ and its DFT (magnitude and phase), all in the same figure using subplots.

For DFT magnitude, normalize (divide by maximum) and plot in decibel scale. Use the command `ylim()` to restrict range between [-100, 10].

Is the phase linear?

(c) Repeat (a) and (b) using Blackman window for $w[n]$ to obtain another filter. Use the command `blackman` to get the window.

(d) Compare and comment on the transition bands and side-lobe levels of the two filters.

(e) Generate 201 samples of the signal $x[n] = \cos\left(\frac{\pi n}{16}\right) + 0.25 \sin\left(\frac{\pi n}{2}\right)$. Filter this signal using the two filters using the `conv()` command. In a single figure, plot the original signal and the filtered signals.

Repeat for the signal $x_1[n] = \cos\left(\frac{\pi n}{16}\right) + 0.25\ \mathrm{randn}\ (1, 201)$.

(f) From the filter $h[n]$ in (a), construct a new filter $h_1[n] = (-1)^n h[n]$. Repeat (b) for this filter and comment on the nature of this filter.

10.2 Digital bandlimited differentiator

[Read Example 6.2.2 from the Proakis & Manolakis textbook and/or Section 7.4.1 of OWN]

A digital differentiator can be designed for digital processing of bandlimited signals sampled at the rate of $f_s = 1\backslash T$. The frequency response of the ideal digital bandlimited differentiator is given by $H_{\text{Diff}}(e^{j\omega}) = j\frac{\omega}{T}$ for $\omega \in [-\pi, \pi]$ where T is the sampling interval.

(a) Show that the impulse response of this filter is

$$h_{\text{Diff}}[n] = \begin{cases} 0, & n = 0 \\ \dfrac{\cos \pi n}{nT} = \dfrac{(-1)^n}{nT}, & n \neq 0 \end{cases}$$

(b) Though ideal, this is a non-causal and un-stable filter with infinite impulse response. Apply the method of windows to obtain FIR filters of length N = 3 and N = 21.

(c) Consider the continuous-time signal $x_c(t) = \sin(2\pi t)$ sampled at $f_s = 100Hz$. Apply the FIR digital differentiators from (b) to perform differentiation. Reconstruct the signal using linear interpolation. Use `t_fine = 0:0.001:3` as the time-grid for representing continuous time-signals (as done in Lab 8). Plot the original signal as well as its derivative. Compare the accuracy of the derivative for different filter lengths.

10.3 Filter design using filterDesigner GUI

This exercise is meant to familiarize you with the filter design tool available in Matlab. Type `filterDesigner` in the command prompt to open its graphical interface. Explore the various FIR & IIR filter design parameters available and relate them to the parameters discussed in the class.

(a) Choose settings to design the exact filter in 10.1 (i.e. low-pass FIR filter using window method) and compare with your answer in 10.1. Note that for an N-length filter, chose filter-order to be N-1.

(b) Navigate the Toolbar at the top of the interface to visualize magnitude response, phase response, impulse response, pole-zero plots, etc.

(c) Design a low-pass FIR Equiripple filter with the following specifications: Passband attenuation 0.5 dB, Stopband attenuation 90 dB, normalized passband frequency of 0.25 normalized stopband frequency 0.5.

(d) Change the design method to least-squares and compare the obtained filter with the one obtained using Equiripple method.

(e) Design an IIR filter with same specifications as given in part (c). Compare the phase response of these filters. What is the difference in the order of the FIR & IIR filters? Which filter requires fewer computations to implement?

10.4 Notch filter (Optional)

A notch filter is a special filter which passes most frequencies virtually unchanged but has zero frequency response at given frequency $\omega = \omega_0$. Such filters are useful to remove single frequency noise (e.g. 50 Hz interference due to power-grid) from signals. We will look at two ways to do this.

(a) **(FIR filter)** An easy way to do this is to place two zeros on the unit circle at $e^{\pm j\omega_0}$. This ensures that there is frequency null at $\omega = \pm\omega_0$. This gives the system function

$$H(z) = b_0(1 - e^{j\omega_0}z^{-1})(1 - e^{-j\omega_0}z^{-1})$$

where b_0 is the gain factor for this filter such that $H(1) = 1$.

Use `freqz()` to plot 2001-point frequency response of this filter when $\omega_0 = \frac{\pi}{4}$.

(b) **(IIR filter)** Alternately, along with the two zeros on the unit circle at $e^{\pm j\omega_0}$, we can additionally place two poles at $r_0\, e^{\pm j\omega_0}$, where $r_0 < 1$. This gives the system function

$$H(z) = b_0 \frac{(1 - e^{j\omega_0}z^{-1})(1 - e^{-j\omega_0}z^{-1})}{(1 - r_0e^{j\omega_0}z^{-1})(1 - r_0e^{-j\omega_0}z^{-1})}$$

where b_0 is the gain factor such that $H(1) = 1$.

Plot the filter frequency response when notch is located at $\omega_0 = \frac{\pi}{4}$ and $r_0 = 0.9$.

(c) Comment on the causality and stability of the above two filters.

(d) Use matlab command `fvtool()` to visualize various aspects of the notch filters in (a) and (b) including their magnitude response, phase response, impulse response, pole-zero plots, etc. How does filter change for $r_0 = 0.5$ and $r_0 = 0.99$?

(e) Load the `handel` sound file in matlab by typing `load handel`. Let this signal be $x[n]$. Listen to it using `sound()`. To this signal add a sinusoidal signal $\sin(2\pi f_0 t)$ of same duration and sampling frequency; set $f_0 = 1024\, Hz$. Listen to this modified signal. Apply the two notch filters designed in (a) and (b) to this modified signal using the `filter()` command. Listen to the two filter outputs.

If you are unable to load the `handel` sound file above, instead generate a 2 second white noise signal in Matlab as follows: $x[n] = rand(1, 2 * f_s) - 0.5$ with $f_s = 8192\, Hz$ and proceed as above.

(f) In a 2x2 figure, plot the first 100 samples of the input signal and output signal from the filter. Verify that your filters are working as expected.