01 November 2023

Lab 9 – Digital filter design

9.1 Filtering of periodic signals with LTI systems

A continuous-time periodic signal $x(t)$ has Fourier series (FS) coefficients a_k. If $x(t)$ is input to the LTI system with frequency response $H(\omega)$, what are Fourier series coefficients of the output signal? What about the periodicity of the output signal?

Ideal low pass filter (LPF): let $\omega_c > 0$ be its cut-off frequency. We wish to find FS coefficients b_k, $k = -N: N$, of the output signal when the input signal with coefficients a_k, $k = -N: N$, is passed through an Ideal LPF.

(a) Write a code for the matlab function `B = myLPF(A,w0_FS,wc)` which takes input signal FS coefficents A, frequency of the input periodic signal `w0_FS`, cut-off frequency wc and returns the output signal FS coefficients in the vector B. Note that your code should be written for general N.

Note: the coefficient a_k corresponds frequency $k\omega_0$, use this information while implementing your LPF.

(b) We will now write a matlab script file to visualize input and output signals of the filter. For this purpose we will use function '`x = partialfouriersum(A,T,t)`' from previous lab session.

>> Initialize $\omega_0 = 1$, and FS coefficients A to obtain the signal $x(t) = \cos(t)$

>> Call function `myLPF` with input A and $\omega_c = 2$

>> Use the function `partialfouriersum` to obtain the time domain signals corresponding to A and B and plot them in the same figure. Use the inputs as T = 2π, and t = -2T:0.01:2T

>> What happens when we change cut-off to $\omega_c = 0.5$?

(c) Ideal high pass filter (HPF): let $\omega_c > 0$ be its cut-off frequency.

>> Repeat (a) for an ideal HPF and write matlab function `B = myHPF(A,w0_FS,wc).`

>> Repeat (b) with the ideal LPF replaced by ideal HPF (continue in the same script file).

(d) Non-ideal filter: let the frequency response be $H(w) = \frac{G}{a+w}$, where G and a are positive real constants. What is the nature of this filter?

>> Write a code to implement this non-ideal filter on FS coefficients A as the matlab function `B = NonIdeal(A,w0_FS,G,a).`

>> Repeat (b) with ideal LPF replaced by the non-ideal filter. Use G = 1, a = 1. >> How is the complex-valued nature of the LTI system frequency response manifested in the output signal?

(e) Repeat the script with the input signal as $x(t) = \sin(2t) + \cos(3t)$. Note that you must appropriately modify A, ω_0, and T for this example. For this input set ideal LPF and ideal HPF filter cut-offs to be $\omega_c = 2.5$.

Optional: repeat when A corresponds to FS coefficients of the periodic square wave.

9.2 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 frequency response and its (linear) phase changed version are given by

$$H_{\text{LPF}}(e^{j\omega}) = \begin{cases} 1, & |\omega| \le \frac{\pi}{6} \\ 0, & \frac{\pi}{6} \le |\omega| < \pi \end{cases} \qquad H_d(e^{j\omega}) = \begin{cases} e^{-j\omega n_c}, & |\omega| \le \frac{\pi}{6} \\ 0, & \frac{\pi}{6} \le |\omega| < \pi \end{cases}$$

What are the corresponding impulse responses $h_{\text{LPF}}[n]$ 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 $x[n] = \cos\left(\frac{\pi n}{16}\right) + 0.25\sin\left(\frac{\pi n}{16}\right)$. Pass this signal through the above two filters using `conv()` command. In a single figure, plot the original signal and the filtered signals.

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

(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.

9.3 Notch filter

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

$$H(z) = b_0(1 - e^{jw_0}z^{-1})(1 - e^{-jw_0}z^{-1})$$

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 $w_0 = \frac{\pi}{4}$.

(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

$$H(z) = b_0\ \frac{(1 - e^{jw_0}\, z^{-1})(1 - e^{-jw_0}\, z^{-1})}{(1 - r_0 e^{jw_0}\, z^{-1})(1 - r_0 e^{-jw_0}\, z^{-1})}$$

where b_0 is the gain factor such that $H(1) = 1$.
Plot the filter frequency response when notch is located at $w_0 = \frac{\pi}{4}$ 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.

9.4 Filter design using filterDesigner

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

(a) Choose settings to design the exact filter in 8.1 (a) (i.e. low-pass FIR filter using window method) and compare with your answers in 8.1. 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.