**Lab III Digital Filters**

Deadline: 2024.4.9

### Functions for Filter

A causal filter is represented as .

Refer to the help document (**doc filter**) in MATLAB for specific definitions, where *a*1=1.

* 1. Develop a function, e.g., [**poles**, **zeros**] = **pzmap**(**bz**, **az**), to evaluate the poles and zeros of the filter *H*(*z*) represented in the rational form, and show its **poles** (‘x’) and **zeros** (‘o’) in Z-plane with a unit circle, where the denominator and numerator coefficients are denoted as **az** = [*a*0, *a*1, …, *aN*] and **bz** =[*b*0, *b*1, …, *bM*], respectively. Is this filter system stable or not, why?
  2. Develop a function, e.g., **Hz** = **freqz**(**bz**, **az, *w***), to calculate the frequency response function **Hz** in a Nyquist interval: [-π, π], where ***w*** denotes the frequency vector in radians.
  3. Develop a function, e.g., **y** = **filter** (**bz**, **az**, **x, L**), to calculate the output **y** in the sequence yn = [0, 1, …, **L**-1] using the I/O difference equation method. Under the initial rest condition and the input **x** = [1, 2, 3] in the sequence xn: [0, 1, 2], calculate and show **y** with **L** = 100.
  4. Use **filter** to calculate and show the impulse response function **hn** with **L** = 100*.*

### Design an FIR filter with the Window method

A simple way to design FIR filter is to use window function truncate the impulse response function of an ideal filter, named the window method.

* 1. Design the low-pass FIR filters, the cut-off frequency *ωc*= 0.3π, with the rectangle window, where the window length *N* = -30~30 and -100~100, respectively. Compare their impulse response functions and FRFs (Real part and Imaginary part) in terms of different window length.
  2. Design a **causal** high-pass FIR filters, the cut-off frequency *ωc*= 0.3π, with the rectangle window and Kaiser window with the window length *N* = 0 ~ 59. And then compare their impulse response functions and FRFs (amplitudes and phases) in terms of the window type.
  3. Signal , where , n = -200:399. The sampling frequency *fs* = 1/T = 200. In order to filter out the cosine of 7 and keep the cosine of 24 of *x*[*n*], design a digital high-pass FIR filter. The filter’s overshoot should be less than 5%. Show FRF of the FIR filter.
  4. Use function *filter()* developed in 1.(c) to filter *x* with the filter you designed in (c) and plot the output signal *y*1. Divide *x* into 3 non-overlapping blocks, 200 samples each, and use the convolution method to filter each block and piece the output blocks together to obtain and plot overall output *y*2. Compare *y*1 and *y*2.
  5. Perform DFT on the steady state of *y*1 and *y*2, respectively, and compare it with the DFT result of *x* to verify the effectiveness of your filter. FFT should be utilized to calculate DFT. Calculate the amplitude of the remaining frequency of 24 according to the length of your filter.

### 3. Design an IIR filter with bilinear transform and realize filtering

The circuit diagram as an analog filter is shown in Fig.1.

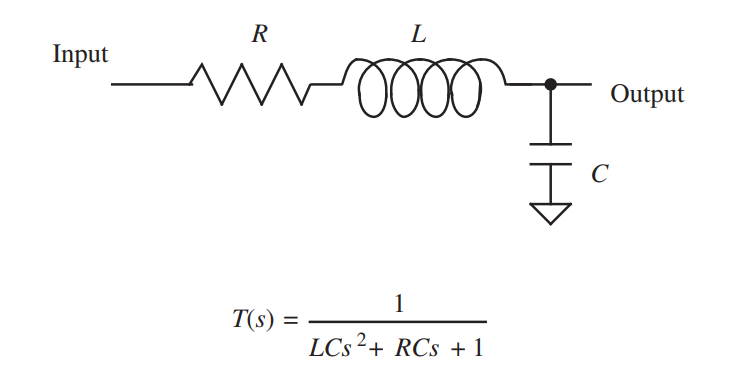


Fig.1 A circuit diagram.

1. Write the transfer function *H*(*s*) of the circuit in terms of *R*, *L* and *C*.
2. Set the sampling frequency *fs*, and write the transfer function IIR filter *H*(*z*) in terms of *R*, *L* and *C* using the bilinear method.
3. Compare the FRFs of the analog filter in (a) and the digital filter in (b), where the circuit parameters *R*, *L* and *C* are 3 Ω, 0.02 H, 0.001 F, respectively. Find the minimum *fs* to satisfy that the difference between the magnitudes of the analog and digital filters is less than 0.1 dB up to 1000 Hz. Show magnitudes in dB and phases in radians, and frequencies from 1 to *fs*/2 in logarithm scale.
4. Produce an input signal *x*[*n*] containing three harmonics *ampltude*@*normalized frequency*, which are 1@5/*fs*, 1@50/*fs* and 1@500/*fs*. Filter the signal to obtain the output signal *y*[*n*] by the IIR filter in (c). Verify the filter effects by comparing the spectra of the input and output signals in their steady-state condition.

### 4. Pole/Zero Designs

Pole/zero placement can be used to design simple filters, such as the notch and comb filters.

* 1. In order to filter out the cosine of 24 and keep the cosine of 7 of *x*[*n*] in 2. (a), we need to design a notch filter with a bandwidth of 1 Hz (Try-and-error way). Show *H*1(*z*), and plot a map of poles and zeroes with the unit circle for reference. Then show FRFs of the notch filter.
  2. Filter *x* with the notch filter and produce the output signal *y*3. Show the time sequences and frequency spectra of *y*3.
  3. In order to filter out the cosine of 7 and keep the cosine of 24 of *x*[*n*] in 2. (c), we can alternatively design a comb filter to keep only 24 of *x* with a bandwidth of 1 Hz. Show *H*2(*z*) and plot a map of poles and zeroes with the unit circle. Then show FRFs of the comb filter.
  4. Filter *x* with the comb filter and obtain the filtered signal *y*4. Show the time sequences and frequency spectra of *y*4.