**Lab II Digital Filters**

Deadline: 6th Nov.

### Design an FIR filter with the Frequency sampling method

The Frequency Sampling Method is to design Finite Impulse Response (FIR) filters by directly sampling the desired frequency response function (FRF) and then using the Inverse Discrete Fourier Transform (IDFT) to compute the time-domain impulse response function (IRF). Normally, require to add a smooth window to suppress fluctuations in FRF.

* 1. Design low-pass FIR filters, where the cut-off frequency is *ωc*= 0.2π, the pass-band magnitude equals 2, the stop-band magnitude equals 0.1, and the phases are zeros. Set the numbers of the frequency samples *N*1 = 10 and *N*­2­­ = 20. Compare the corresponding impulse response functions *h*1[n] and *h*2 [n].
  2. Develop a function to evaluate FRF, i.e., H = my\_freqz(b, a, N), where ‘b’ and ‘a’ are coefficient vectors of the numerator and denominator polynomial functions of the system function, *N* is the number of frequencies over the whole Nyquist range, and H is the corresponding FRF. Use this function to calculate FRFs (H1 for *h*1) and (H2 for *h*2) and compare their magnitudes and phases.
  3. Design **causal** high-pass FIR filters, the cut-off frequency *ωc*= 0.5π, with the rectangle window (denoted as H3) and Kaiser window (H4) with the same window length *L =* 20. Compare their IRF and FRFs, respectively.
  4. Signal , where , n = -500: 499. The sampling frequency *fs* = 1/T = 100. In order to filter out the harmonic at 5 and keep that at 5.5 of *x*[*n*], design a causal high-pass FIR filter H5 using frequency sampling method with order M. The filter’s fluctuations at pass-band should be less than 1%, the fluctuations at stop-band should be less than 1%, and the order M should be as small as possible. Show M and the FRF of H5.
  5. Signal , where , n = -5000: 4999. Divide *x* into 5 non-overlapping continuative blocks (each with the length 2000) and use the convolution method to filter *x* with the filter in (d) with a suitable M, and the whole output is *y*. Compare *x* and *y*. Point out the steady-state output *y*1.
  6. Compare spectra of *x* and *y*1 and verify the effectiveness of the filter.

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

1. Signal , where , *n* = 0: 99. In order to filter out the cosine of *f2* and keep *f1*, we need to design a digital low-pass filter using bilinear transform. The prototype filter is a second-order Chebyshev2 analogy filter, where *fc* = 0.2. Show ***az***and ***bz*** of the filter and plot FRF.
2. Develop a sub-function *y* = **myfilter** (bz, az, *x*) by using the I/O difference equation method.
3. Use **myfilter** to filter *x*[*n*] and produce the output signal *y*. Compare the time sequences and frequency spectra of *x* and *y*, respectively.
   1. **Pole/Zero Designs**

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

* 1. In order to only filter out the harmonic at 5 in *x*[*n*] in 1(d), we need to design a notch filter with the try-and-error way. Show poles and zeroes in a plot. Then show FRFs of this notch filter.
  2. Filter *x* with the notch filter and produce the output signal *y*1. Calculate the amplitudes of harmonics at 5 and 5.5 in *x* and *y*1, respectively.
  3. In order to enhance the amplitude at 5 with 2 times and keep other frequencies unchanged, we can design a comb filter with a small bandwidth at 5, e.g. less than 0.2 Hz. The bandwidth can be evaluated as the band width at half power of the peak (or 3 dB down). Show K(scale), P(Poles), Z(Zeros) of this filter. Then show FRFs of the comb filter.
  4. Filter *x* with the notch filter and produce the output signal *y*2. Calculate the amplitudes of harmonics at 5 and 5.5 in *x* and *y*2, respectively.