IIR Filter Design

- Design a Butterworth digital IIR lowpass filter using impulse invariant transformation by taking T = 1second to satisfy the given specifications.

- Design a Chebyshev digital IIR lowpass filter using bilinear transformation by taking T = 1second, to satisfy the given specifications

• Specification:

Gain at passband edge frequency

Gain at stopband edge frequency

Passband edge digital frequency

Stopband edge digital frequency

Sampling time

Passband attenuation in dB

Stopband attenuation in dB

Butterworth filter

- buttord calculates the minimum order of a digital or analog Butterworth filter required to meet a set of filter design specifications.
- [n,Wn] = buttord(Wp,Ws,Rp,Rs) returns the lowest order, n, of the digital Butterworth filter with no more than Rp dB of passband ripple and at least Rs dB of attenuation in the stopband. The scalar (or vector) of corresponding cutoff frequencies, Wn, is also returned. Use the output arguments n and Wn in butter.
- [b,a] = butter(n,Wn) returns the transfer function coefficients of an nth-order lowpass digital Butterworth filter with normalized cutoff frequency Wn.

Chebyshev Filter

- ## Cheb1ord: Chebyshev Type I filter order

[n,Wp] = cheb1ord(Wp,Ws,Rp,Rs) returns the lowest order n of the Chebyshev Type I filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband.
The scalar (or vector) of corresponding cutoff frequencies Wp, is also returned.
Use the output arguments n and Wp with the cheby1 function.

- ## Cheby1 : Chebyshev Type I filter design

[b,a] = cheby1(n,Rp,Wp) returns the transfer function coefficients of an nth-order lowpass digital Chebyshev Type I filter with normalized passband edge frequency Wp and Rp decibels of peak-to-peak passband ripple.

Bilinear transformation method for analog-to-digital filter conversion

[num ,den] = bilinear(num,den,fs) convert an s-domain transfer function given by num and den to a discrete equivalent.

Impulse invariance method for analog-to-digital filter conversion

[bz,az] = impinvar(b,a,fs) creates a digital filter with numerator and denominator coefficients bz and az, respectively, whose impulse response is equal to the impulse response of the analog filter with coefficients b and a, scaled by 1/fs.

Digital Transfer Function is,

Hz = tf (num, den, T)

To get the frequency response

H = freqz(...,W) returns the frequency response at frequencies designated in vector W, in radians/sample (normally between 0 and pi).

Then get magnitude response